Universidad
Carlos III de Madrid

Departamento de Ingeniería Electrónica

TRABAJO FIN DE GRADO

# Sistema empotrado para reconocimiento de firma manuscrita

*Autor:* Pablo Díez López

*Tutor:* Luís Mengíbar Pozo

Madrid, Septiembre 2015

**Título:** Sistema empotrado para reconocimiento de firma manuscrita
**Autor:** Pablo Díez López
**Tutor:** Luís Mengiíbar Pozo

# EL TRIBUNAL

Presidente:

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día ..... de ..... de ... en ....., en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

VOCAL

SECRETARIO PRESIDENTE

# Agradecimientos

*Este apartado es, sin duda, el último que se escribe, el primero que se coloca y probablemente, el último que se lee.*

Quiero agradecer en primer lugar a mi familia. A mis hermanos Manuel y Gustavo. Por su apoyo constante, por enseñarme que nunca hay que darse por vencido y que siempre hay que luchar por conseguir lo que se desea. Lo difícil no es conseguirlo, sino saber lo que uno quiere. Porque ellos han sido como unos segundos padres.

A mis padres reales, por la educación que me han proporcionado, por darme la oportunidad de estudiar estos años en Madrid y alrededores, por su confianza y ayuda en cada momento. Sin la menor duda esta estapa ha sido determinante en mi crecimiento personal. Sin ellos nada de esto habría sido posible.

Agradecer también a la Universidad Carlos III por las facilidades que me ha brindado para complementar mi formación en otras universidades de prestigio internacional. Estos intercambios me han aportado amistades que durarán para siempre, especialmente a Marta, una pieza clave tanto en Praga como posteriormente, por la oportunidades que me brinda.

Asimismo, me gustaría mostrar mi gratitud a la organización BEST por enseñarme diferentes aptitudes personales y del mundo profesional y "humanizar" el campus de Leganés.

Posteriormente, quiero dar las gracias a mi tutor, Luís Mengíbar, por dirigirme este proyecto y sobre todo darme flexibilidad sobre las decisiónes del diseño. También por todos sus consejos y por asesorarme siempre que lo he necesitado. Porque gracias a él he conseguido sacar adelante este trabajo y extraer los máximos conocimientos del mismo.

Seguidamente, quiero dar las gracias a Pablo e Isa, mi fieles compañeros. Por celebrar conmigo los buenos momentos y limpiarme las gafas cuando todo estaba borroso durante la carrera. ÀYa la he terminado?

# Abstract

Society is evolving and technology is making this evolve enforceable. Some examples of it are: the way people communicate, the working methods, or the processes to identify.

The science of biometrics is based on discovering the identities of human. The way it is done, is by investigating their physical or behavioral traits. Signature belong to behavioral traits. It has been used for last 5 centuries to give authorship of a work or to accept an agreement. Nowadays it is still the most accepted biometric recognition technique.

This project will create a functional identification system using a embedded processor. The trait under study will be the signature.

**Keywords:** Raspberry Pi,signature recognition, DTW, Wacom.

# Summary

Biometrics identification systems are becoming more used. Nowadays it is common to find biometric sensors in technology devices such as smart phones and computers. These systems are designed to verify the identity, not to identify it.

## 0.1. Definition of the project

The aim of this project is to develop a prototype of a signature recognition system with a embedded processor. The prototype will prove the possibility of using embedded processors for biometric identification (not only verify).

The system will be designed for the final user. The interface will be as simple as possible but also the system delay should be low to prevent rejection of the device. To reach this low computation time, different algorithms will be under study, and also different improvements.

As the device should be a functional prototype, a low error rate should be reached. During the developing, the software has been optimized for the chosen hardware, lowering the error rate until achieving good values.

## 0.2. Algorithm and hardware election

To succeed in this project, different hardwares and algorithms have been investigated. Following, these concepts will be explained in more detail.

**Hardware**

There is various possibilities of embedded processors in the market. The one chosen was the Raspberry Pi 1 model B+ for different reasons. This board gives high connectivity as well as many possibilities of expansion in the future. Also Unix operating systems can be installed in the hardware. This is also one of the cheapest boards.

Raspberry Pi is build with a Broadcom chip that contain a ARM 11 at 700MHz CPU (Central Processor Unit), 512 MBytes of SDRAM memory with 400MHz of clock and a Broadcom VideoCore IV at 250 MHz as GPU (Graphics Processor Unit).

The board has 17 GPIO (General Purpose Input Output), 1 jack audio output, 4x USB 2.0, 1x 10/100Mbit/s Ethernet (RJ45),1 HDMI 1.4, 1 microSD slot ( maximun 32 GB)

The other options for hardware were: Intel Edison, BeagleBone (Texas Instruments), MicroBlaze, Arduino, NucleoST (STM Microelectronics), Xilinx Zyinq-7000 and Nvidia Jetson TK1.

The other hardware piece needed for this project was a **caption device** where the user could perform the signature. The device chosen was a tablet Wacom Intuos 4 PTK-440 because it has a big sensitive area (157.5 X 98.4 mm), high resolution in this area (200 lines per mm) and high pressure resolution (2048 points). It also measure the position of the pen with 2 angles (azimuth and inclination).

**Algorithm**

There are different methods to recognize signatures. Some of the methods that where studied to implement this project where HMM(Hide Markov Models), GMM (Gaussian Mixture Model), ANNs (Artificial Neural Networks) , SVMs (Support Vector Machines) and DTW (Dynamic Time Warping). The last one, DTW , was the algorithm chosen because it has really good results measuring similarity between two temporal sequences.

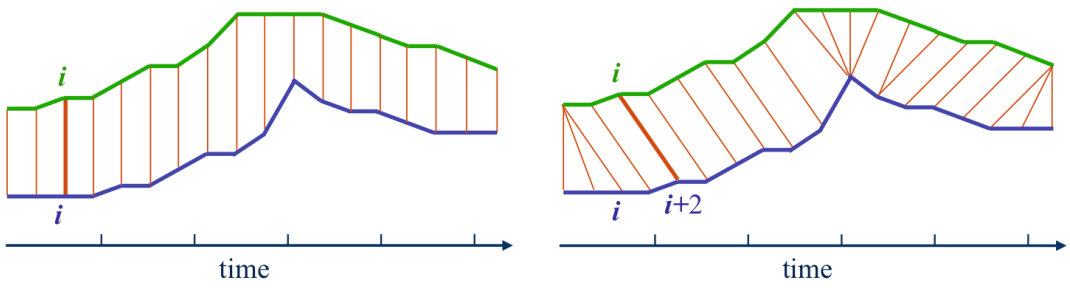



Figura 1: Left:2 sequences in time without DTW comparation. Right: How the DTW align the similar points of the sequences

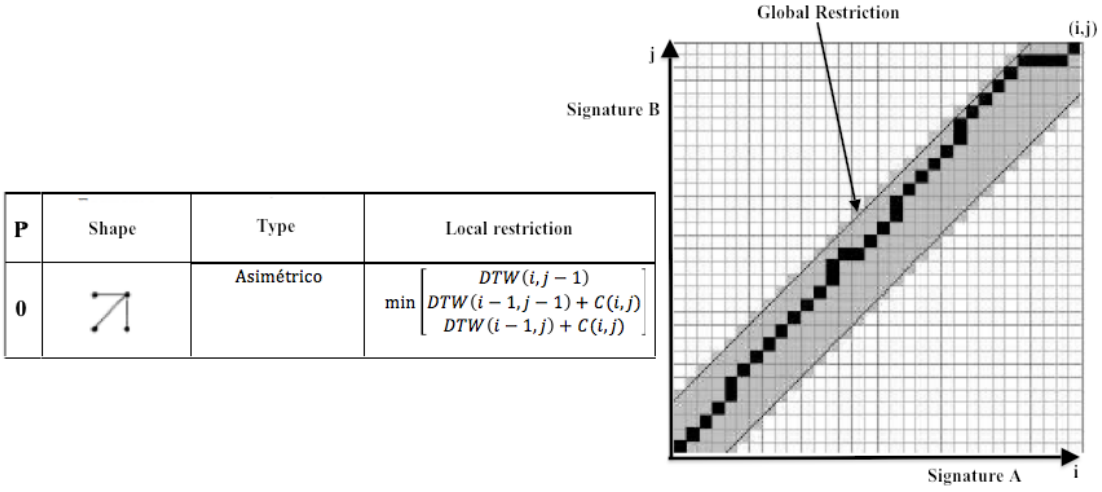
In the figure 1 it is shown how the DTW algorithm aligns the 2 time series of data.

This algorithm has been adjusted according with the best parameters studied in other papers. The local restriction chosen was P(0) asymmetric and the local cost was the euclidean distance, (visible in figure 2).

To reduce the computation costs, some global limits were added to the DTW matrix calculation. This function reduces the area where DTW can align the 2 signatures. The function used was the Sakoe window. It is defined in the right part of figure 2. The window limits the values that the DTW can align. The width of the windows was adjust at 30 points.

## 0.3. Software development

This section explains how the software works. The system has been implemented in the language C and it runs over Raspbian OS. This operating system is a




| P | Shape | Type | Local restriction |
|---|---|---|---|
| 0 | | Asimétrico | $\min \begin{bmatrix} DTW(i,j-1) \\ DTW(i-1,j-1)+C(i,j) \\ DTW(i-1,j)+C(i,j) \end{bmatrix}$ |

Figura 2: Left: local restriction Right: global restriction, Sakoe Window

modify Debian (Linux) for Raspberry Pi.

The software has been develop in modules, giving option to change some of them easily in future work. It is also useful in case that the acquisition tablet changes because you would only need to change the data acquisition module.

- **Data acquisition**

  The input data (signature) of the system comes from the Wacom tablet. This information is delivered to the program through USB connection. It has been one of the hardest modules to develop because the usual way to read data from an Interrupt type USB connection is through the endpoint address and, unfortunately, this does not work on Raspberry Pi.

  At the end the data is read from the USB using a developing module from the linux kernel called **usbmon** that is part of **debugfs**. This module is for the USB what TCPdump is for network connection. It can read all the information passing through USB ports.

  The acquisition module get the information coming from the Wacom tablet reading buffer that usbmon creates.

  There is a second acquisition module. This is used only to add users from an already built database. This module is only active during the algorithm training period and it is used to adjust the system. The data is obtain from MCyT database. This database has 100 users with 25 real and 25 fake signatures per user.

- **Database**

  It is essential to have a database in a identification system. In the database all the signatures (5 per registered user) are stored. The reason to store 5 signatures per user is to have the variability of the signature.

  All the signatures are saved in a subdirectory called “db” inside the program directory. The program creates one file per signature and it is identified by

the numbers located on the name of the file.

- **Preprocessor module**

  Raw data flows from the acquisition module. The aim of this module is to transform the raw data to improve the results and reduce the DTW comparison time.

  All the signatures pass through this process and these are the steps:

  1. Geometric normalization: The coordinates (x,y) of the signature are modified according to the equation 1. This moves the coordinate axis of each signature to the mass center of the signature. This transformation allows the user to sign in any part of the sensitive area of the tablet.

$$\begin{aligned} y_t^N &= (y_t - \mu_y) \\ x_t^N &= (x_t - \mu_x) \\ \mu_k &= (\sum_{i=1}^{N} f_{k,i})/N \\ k &= x, y \end{aligned} \tag{1}$$

  2. Linear interpolation: all the signatures are resampled to the same number of points. The signatures with less points than the sampling number, will keep the same number of points, there will be no interpolation for them. There is two reasons to make the interpolation. The first one is that DTW works better with same length signatures.(we concluded this after making some experiments with DTW algorithm). The second one is that the computation time is directly proportional to the sampling number. To have a fast system, the sampler number should be as low as it can reach.

  3. Calculation of evaluated characteristics: on the one hand the datas obtained from the tablet are coordinate (x,y), pressure (p), azimuth and inclination angles. On the other hand, DTW algorithm is evaluating the speeds (dx,dy), pressure (p) and coordinate (y). To fill this gap, the formulas written in equation 2 are used after the interpolation.

$$\begin{aligned} S &= y, dy, dx, p \\ dx &= (x_{t+1} - x_t)/\Delta t \\ dy &= (y_{t+1} - y_t)/\Delta t \end{aligned} \tag{2}$$

- **Algorithm DTW**

  This is the central module of the system. It creates the data that the classification module will evaluate later on. It evaluates the differences between the input signature and all the signatures in the database. The DTW rates every pair of signatures giving a value according to the restrictions explained previously.

  The return value from this module correspond to the DTW value between 1 signature and 1 user (5 signatures). This value comes form the arithmetic

average of the 5 DTW values. The values are achieve from the comparison of the input signature (A) with the 5 signatures of the same user which stored in the database.

The signature characteristics that the DTW evaluates are: speed on coordinate X (dx), speed on coordinate Y (dy), pressure (p) and coordiante (y). This are the most significant features according toÊ[9].

$$DTW_{signature_A,user_N} = \frac{1}{5}\sum_{i=1}^{5} DTW_{signature_A,signature_i} \tag{3}$$

- **Threshold calculation**

  The system has been designed with a global threshold value. This value is adjusted during the design period. This part of the program does not run during identification mode.

  The way to calculate the threshold is to find the EER point (Equal Error Rate). This point is located when the FAR skilled (False Acceptance Range) and FRR (False Reject Range) has the same value.

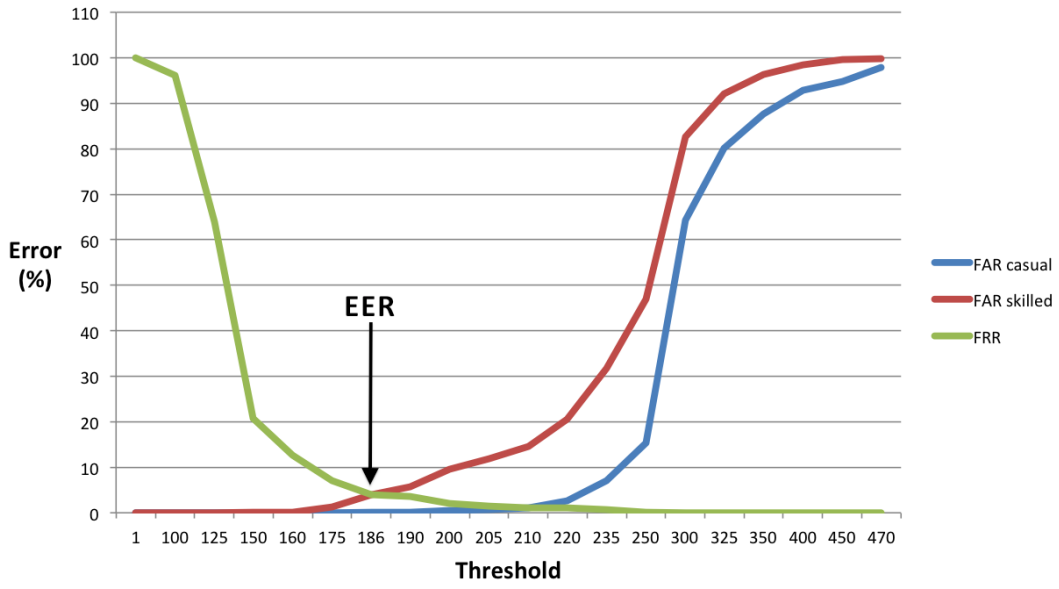



Figura 3: Threshold calculation.This represents the error you get form all the range of thresholds.

- **Improvements**

  There is some improvements implemented in this system. One of it reduces the computation time and the other reduce the error.

  1. Time difference filter: this filter checks the longitudes (in time) of the 2 signatures. If the time difference is too high, DTW value is not needed to be calculated, for this reason we assigned a really high DTW. This value is much higher than the threshold, so the signature is not accepted. It reduces 6 % the computation time.

2. Starting speed: This part evaluates the initial speed of the signatures. This initial speed is highly different between true and forged signatures. If the speeds are too different the final DTW value increases. This reduces the error from 5.5 % to 4.7 % (EER values).

## 0.4. Final results and Conclusions

The system has been optimized to work in this processor. These are the final parameters of the system:

- Sampling: 50 points
- Threshold: 186

The final results that define the system are the computation time and the error. The computation time is the time it takes to compare 1 signature with 1 user (5 signatures).

- False Acceptance Range Skilled: 4 %
- False Acceptance Range Casual: 0.1 %
- False Reject Range: 4 %
- Computation time: 11ms (per user in the database)

We can reach a database of 350 users and the computation time will be 4 seconds. This data is really good with a 700 MHz processor.

In conclusion, we have develop a prototype of a signature identification system with a Raspberry Pi 1 and a tablet Wacom. The error reached from the system is 4 % being acceptable for a prototype. All the aims have been reached.

# Índice general

# Índice de figuras

# Capítulo 1

# Introducción

Todos vemos en nuestro entorno el avance tecnológico que la sociedad ha experimentado en las ultimas dos décadas y tambien su indudable informatización.

Este avance ha creado una tendencia a utilizar cada vez mas la tecnología para tareas que anteriormente se realizaba sin la misma. Estas tareas o procedimientos abarcan un amplio abanico, entre ellas: la comunicación, la forma de trabajar o el aprendizaje.

Los métodos de identificación tambien han evolucionado, principalmente por razones de privacidad y seguridad. Actualmente tenemos mas conciencia sobre la importancia de los datos personales, ya que, con estos datos se puede llegar incluso a suplantar la identidad. Por ello las seguridad que los sistemas implementan se van endureciendo.

Podemos ver los cambios en las medidas de seguridad en métodos de identificación como, por ejemplo, ciertos requisitos que tiene que cumplir algunas contraseñas (contener mayúsculas, minúsculas y, al menos, un número) o la verficación de dos pasos de Google, en la que ademas de la contraseña, hay que escribir un código que Google envía en ese momento a tu teléfono movil.

## 1.1. Motivación del proyecto y objetivos

El principal objetivo de este proyecto es demostrar la viabilidad de utilizar microprocesadores empotrados para la identificación biométrica.

Queremos construir un sistema funcional que identifique, o al menos verifique, la identidad de un usuario por medio de la firma manuscrita. Se buscará distintas posibilidades dentro del hardware y software que más se adapten tanto a el objetivo como al presupuesto.

En este desarrollo se tendrá en cuenta el usuario final, que es el que lo utilizará . Para evitar rechazos al sistema, se tendrá en cuenta tanto la sencillez de utilización como los tiempos de espera.

Con este proyecto se quiere demostrar la utilidad de algunos conocimientos adquiridos durante la carrera y la capacidad para analizar y tomar decisiones de diseño. Se necesitan conocimientos de múltiples campos como electrónica, programación, estadística o teoría de la detección

## 1.2. Estructura del documento

A continuación y para facilitar la lectura del documento, se detalla el contenido de cada capítulo.

- En el capítulo 1 se realiza una introducción, marcando los objetivos y motivaciones del proyecto.
- En el capítulo 2 se hace un repaso todos los conceptos necesarios para el buen entendimiento de este trabajo.
- En el capítulo 3 se explica el desarrollo del sistema, tanto su funcionamiento como la forma en la que se ha implementado.
- En el capítulo 4 se exponen las conclusiones.
- En el capítulo 5 se estudian las distintas opciones para seguir desarrollando este proyecto.
- En el capítulo 6 se detalla el presupuesto.
- En el capítulo 7 se muestra la bibliografía.
- En el Anexo 1 se explica la planificación que se ha seguido.

# Capítulo 2

# Estado del arte

La función de este capítulo es dar a conocer todos los conceptos necesarios para entender correctamente el presente trabajo.

Se empezará por la idea de biometría, explicando los diferentes procedimientos de identificación que podemos encontrar tanto en el mercado como los que se están desarrollando. La firma manuscrita será el método donde haremos más incapié ya que es el cual trata este trabajo. El algoritmo utilizado para la identificación será el DTW, Dynamic Time Warping, por su buen funcionamiento en este campo. El hardware sobre el cual va implementado el proyecto es una Raspberry Pi.

## 2.1. Biometría

También llamada identificación biométrica, es el reconocimiento de los seres humanos mediante sus rasgos físicos o de conducta. La utilización de estos procedimientos sirve para el control de accesos y para la identificaión de personas.

Estos sistemas utilizan rasgos humanos conocidos como identificadores biométricos para el reconocimiento, los cuales, pueden ser físicos o conductales. Entre los primeros podemos encontrar la huella dactilar, el iris o el ADN. Alguno de los identificadores conductales o de comportamiento son la forma de caminar, la voz o la firma [20].

Cada vez se estan adoptando más sistemas biométricos [1] para identificar a los individuos o controlar el acceso de los mismo ya que son más fiables que los sistemas convencionales mediante objetos (llave) o mediante conocimientos (contraseñas, datos personales).

### 2.1.1. Sistemas biométricos: Identificación y verificación

Un sistema biométrico es aquel que esta diseñado para llevar a cabo labores de biometría. Podemos dividirlo en tres grandes partes [8]:

- Adquisición del identificador biométrico.
- Compresión, procesamiento, almacenamiento y comparación de los datos adquiridos con la base de datos.
- Interfaz.

En la figura 2.1 se pueden ver todos los componentes de un sistema biométrico. La "adquisición del identificador biométrico" anteriormente nombrada esta reflejada como "recolección de datos". El resto del partes de la figura corresponden a la parte de "compresión, procesamiento, almacenamiento y comparación de los datos adquiridos con la base de datos".

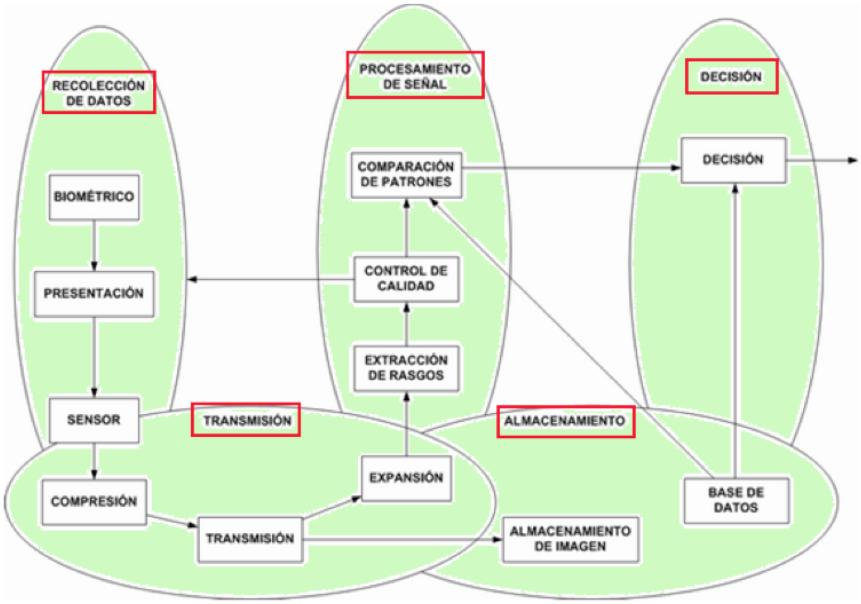



Figura 2.1: Esquema de un sistema biométrico genérico [8]

Dentro de los sistemas biométricos podemos diferenciar dos tipos según su funcionamiento:

- Sistemas de identificación.
- Sistemas de verificación.

A continuación se describen cada uno de ellos con más detalle. Ambos tipos comparten el esquema interno mostrado en la figura 2.1.

### Sistemas de identificación

Responden a la pregunta "¿Quién es usted?".Tratan de identificar un sujeto por medio de una o más características biométricas. Una vez tenidas estas características, se hace una comparacion de las mismas con todos los usuarios de nuestra base de datos (1:N). Para su correcto funcionamiento el sujeto en cuestión debe estar previamente registrado, figurar en la base de datos.

### Sistemas de verificación

Estos sistemas responden a la pregunta "¿Es usted quien dice ser?". Comprueban la identidad del individuo comparando las caraterísticas recogidas con únicamente los datos que tenemos guardados de ese individuo en la base de datos. La comparación es 1:1 y la salida del sistema es correcta (la identidad del usuario es verdadera) o incorrecta.

Para caracterizar un sistema de este tipo utilizamos dos conceptos que giran en torno a la fiabilidad.

- **Tasa de falsa aceptación** (FAR, False Acceptance Rate): es la proporción de identificar un usuario no legítimo respecto de usuarios totales en el test. Básicamente decir que un usuario es"correcto"pero en realidad es "incorrecto"teniendo en cuenta el número total de usuarios comprobados.
- **Tasa de falso rechazo** (FRR, False Reject Rate): es la proporción de los usuarios rechazados cuando el usuario era legitimo respecto del número total de usuarios evaluados.

Estas tasas son inversamente proporcionales y varían según ajustemos el umbral de decisión. El punto de cruce entre la curva FRR y FAR se llama ERR (Equal Error Rate) y es el punto del umbral donde ambos errores son iguales.

Para buscar el umbral óptimo punto EER visible en la figura 2.2, se realiza un barrido desde un umbral muy bajo (poca sensibilidad) hasta un umbral alto (mucha sensibilidad).

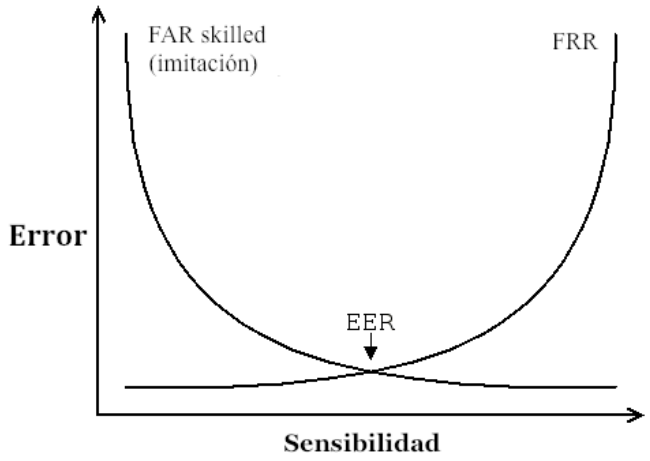



Figura 2.2: Gráfica de las curvas FAR y FRR

Los sistemas que se suelen construir son de verificación (comparación 1:1) ya que el coste computacional de un sistema de identificación puede ser muy elevado ya que es proporcional a la base de datos disponible (comparación 1:N). El sistema que se ha construido puede funcionar tanto en modo identificación como en modo detección.

### 2.1.2. Métodos de identificación biométrica

Podemos dividir los patrones biométricos en dos subclases, los que se basan en las características de comportamiento y los que se basan en las características físicas.

### Métodos basados en características físicas

- **Iris ocular** : Se capta una imagen de alta resolución del iris del individuo. El patrón utilizado, el iris, es muy estable con el paso del tiempo. Sólo en raras ocasiones de traumatismo puede cambiar. Debido a esta estabilidad y a que el método en sí no es muy costoso, es uno de los mejores identificadores biométricos. No hay que confundirlo con el escaneo de la retina que el método es similar, el objeto observado es diferente.

- **Identificación vascular**: Los vasos sanguíneos superficiales son el objeto por el cual identificamos al individuo. Las zonas de estudio de esos vasos son la palma de la mano o un dedo. Este sistema esta en desarrollo todavía y no esta muy extendido.

- **Huella dactilar**: Uno de los sistemas más usados hoy en día. Podemos encontrar estos identificadores en móviles y ordenadores. Debido a su cómoda utilización, actualmente, muchos de estos aparatos se pueden configurar para limitar el acceso con este método.

Figura 2.3: lector de huella dactilar en un ordenador portatil (Fuente: lenovo.com)

- **Geometría de la mano**: Es una de las primeras técnicas implementadas (salió al mercado en los años 80) [15]. Este método analiza diferentes elementos de la mano, en la figura podemos ver un ejemplo de los parámetros utilizados. Las maquinas utilizadas para este cometido eran bastante aparatosas, pero actualmente incluido con la cámara de un smartphone se puede hacer este tipo de identificaión [2]. En la figura 2.4 se puede ver un detector de la geometría de la mano. Los cilindros sirven para la correcta colocación de la mano haciendo al captar los parámetros de la misma , siempre esté en la misma posición.

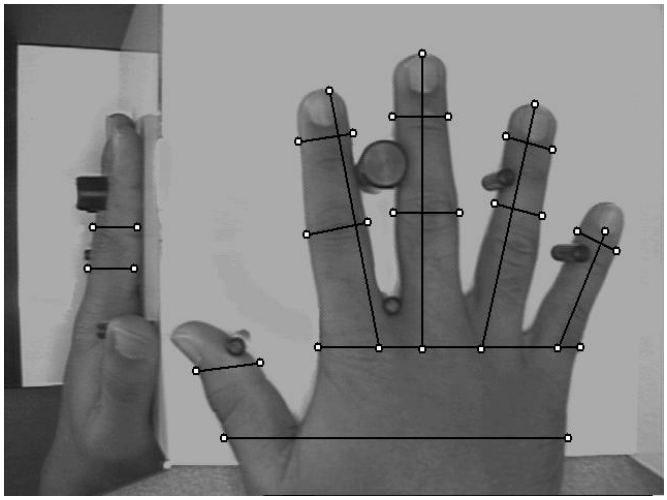


Figura 2.4: Geometría de una mano con los parámetros señalados (Fuente: ibiblio.org)

### Métodos basados en características de comportamiento

- **Dinámica de tecleo**. Se utiliza el parametro dinamico de tecleo del usuario para su autenticación. Posteriormente se compara con las plantillas que el sistema tiene. Este metodo da una confianza de autenticaión del 89 % [12].

Figura 2.5: Dinámica de tecleo (Fuente: appliedbiometrics.com)

- **Firma**: También llamada rúbrica, probablemente el metodo de identificación más extendido y generalizado. Actualemente tiene fines identificatorios, jurídicos, representativos y diplomaáticos. Es el método sobre el cual gira este trabajo.

Figura 2.6: Tableta de la marca Wacom para adquirir la firma(Fuente: wacom.eu)

- **Escritura manuscrita**: Utilizada ampliamente en la historia para verificar la autenticidad de documentos. Los especialistas en esta materia analizan los rasgos característicos del documento a compara con uno del que se sabe la autoría.

- **Voz**: Debido a la facilidad de tomar muestras (con un micrófono), suele ser la elección para la identificación de individuos remotamente. Estos métodos se empezaron a desarrollar desde 1970 con filtros analógicos hasta hoy en día que la NSA (National Security Agency) y el NIST (Instituto Nacional de de Estándares y Tecnología) tienen la tecnología más desarollada [11].

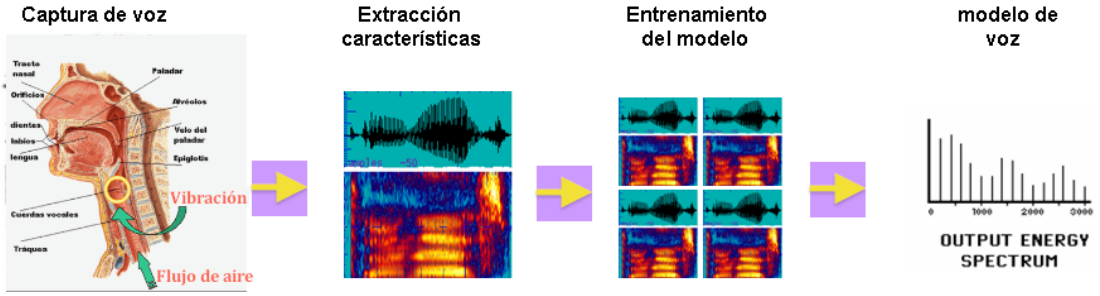



Figura 2.7: Pasos del reconocimiento de voz (Fuente: monografias.com)

### 2.1.3. Comparativa de los sistemas

Actualmente no existe un sistema que sea el mejor. La elección del mismo dependerá de distintos factores. Los más relevantes son el modo de funcionamiento del mismo (identificación o reconocimiento), el número de usuarios de la base de datos, la localización, los riesgos de seguridad, el tipo de publico que vaya a tener.

Como se puede apreciar en la figura 2.8 , la huella dactilar es el método biométrico más utilizado con el 58 % y la firma esta en tercer lugar con el 12 %.

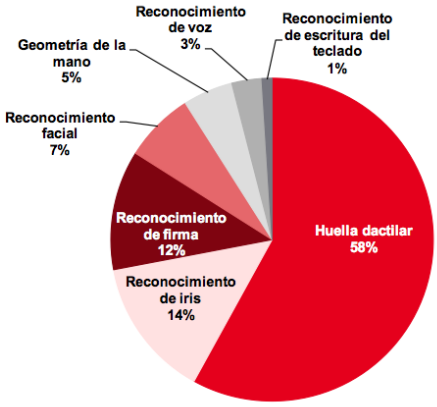



Figura 2.8: Grado de implantación en el mercado (Fuente: Central instituto of Technology of Kokrajhar 2010)

En la figura 2.9 se muestra una tabla comparativa de los principales sistemas biométricos respecto de diferentes características. Entre ellas podemos encontrar las que tiene que ver con la sociedad ("Grado de aceptación"), con el rendimiento ("Resistencia al fraude") o las que miden como se comporta a lo largo del tiempo ese tipo de método biométrico("Permanencia"). En la tabla; "A" es alto, "M" medio"B" es bajo.

| | Grado de Aceptación | Resistencia al fraude | Mensurabilidad | Permanencia | Unicidad | Universalidad |
|---|---|---|---|---|---|---|
| Huella dactilar | M | A | M | A | A | M |
| Reconocimiento facial | A | B | A | M | B | A |
| Reconocimiento de iris | B | A | M | A | A | A |
| Geometría de la mano | M | M | A | M | M | M |
| Reconocimiento de retina | B | A | B | A | A | A |
| Geometría de venas | M | A | M | M | M | M |
| Reconocimiento de voz | A | B | M | B | B | M |
| Reconocimiento de firma | A | B | A | B | B | B |
| Reconocimiento de escritura de teclado | M | M | M | B | B | B |
| Forma de andar | A | M | A | B | B | M |

*A: Alto M: Medio B: Bajo*

Figura 2.9: Valoración comparativa de las distintas técnicas biométricas [16]

### 2.1.4. Inconvenientes de la biometría

La biometría no esta libre de problemas. Algunos de estos son de ámbito ético como la privacidad, otros son de ámbito tecnológico como el coste computacional. Este campo es un tanto delicado ya que los sistemas biométricos interactúan directamente con los usuarios.

#### Privacidad de la información

Los datos que los diferentes sensores biométricos recogen son datos directamente relacionados con la persona. Algunos de estos métodos obtienen información extra que no es necesaria para la identificación pero que puede violar la intimidad. Uno de los ejemplos más polémicos y claros se encuentra en el mejor metodo biométrico, el identificador de ADN. Una muestra original contiene, además del ADN, información que puede dañar al individuo si se hiciera pública como la tendencia a tener una enfermedad.

### Privacidad corporal

Los métodos de identificación biométrica pueden clasificarse también teniendo en cuenta si son o no invasivos. Los invasivos; olor, ADN, retina ,electrocardiograma; necesitan un contacto directo con el cuerpo del individuo. Este contacto produce rechazo por parte de la sociedad. Incluso el lector de huella puede violar la privacidad corporal para aquellos que defienden el contacto físico como no saludable.

Otros métodos no invasivos, o al menos, menos invasivos son la dinámica del tecleo, la voz o la firma, en general, más aceptados por los individuos.

### Integridad del individuo

Al inventar o desarrollar una tecnología, sistema o metodología siempre hay que pensar a la inversa y encontrar las formas de "hackearlo". De esta forma se puede seguir desarrollando el sistema para que sea más seguro. En este campo, cada método de lectura biométrica lleva consigo una forma de "hackearlo" . Alguna de estas formas pueden dañar la integridad del individuo.

Poniendo un ejemplo un tanto exagerado, en una puerta controlada con lector de huella, un ladrón puede cortarte el dedo al usuario legitimo para poder tener acceso. Para este mismo fin, si se utiliza una llave convencional, el simple robo de la misma valdría, evitándonos así quedarnos sin un dedo.

Dado que este trabajo implementa un sistema de reconocimiento de firma, no existe riesgo de dañar la identidad del individuo.

### Identificadores Irremplazables

La unicidad de los rasgos biométricos es a la vez lo que da fortaleza a este tipo de indentificación y también riesgo a quedarte sin acceso. Si el rasgo en cuestion se ve alterado, ya sea por una enfermedad o por una agresión podria dejar de ser válido. La huella dactilar es el identificador biométrico más usado [20] y también uno de los más expuestos. Esta parte del cuerpo la usamos a diario y para ejecutar acciones ya que se encuentra en las manos. Nuestra huella puede variar debido en el más común de los casos a un corte y su posterior cicatriz.

### Coste computacional

El último paso dentro del proceso de indentificación es la computación por medio de un algoritmo de los datos obtenidos por medio de el sensor o sensores correspondientes. El coste de comparar una muestra con otra muestra (verificación, 1:1) no es grande, pero al comparar una muestra con una base de datos (identificacion, 1:N), el coste se dispara.

## 2.2. Firma como identificador biométrico

Después de contextualizar la biometría exponiendo los métodos más extendidos hoy en día, este apartado se centra en el que este proyecto concierne, la firma. Hay dos ramas dentro del análisis de la firma que se rigen por los datos que tenemos disponibles sobre la misma. Reconocimiento off-line y on-line. Este trabajo trata sobre el procedimiento on-line.(Estos conceptos se explican en el apartado 2.2.2)

### 2.2.1. Caracteristicas de la firma

- **No invasora**: Es un procedimiento en el que el cuerpo del sujeto no interactúa de manera directa con el sensor. En nuestro caso se interactuará con el bolígrafo propio de la tableta gráfica.

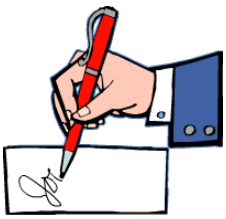


Figura 2.10: firma manuscrita

- **Ejecución generalizada**: La acción que esta lectura bioétrica implica, es bien conocida por la sociedad. Todos hemos firmado en muchas ocasiones. No es necesario ningún aprendizaje previo para realizar la prueba.

- **Aceptación social**: Debido a que lleva usándose muchos siglos, tenemos la firma interiorizada como la forma de mostrar conformidad o autoría de un escrito. El acto de hacer una rúbrica no crea conflicto, como por ejemplo en análisis de ADN.

- **verificación manual**: Se pueden comparar dos firmas visualmente y, salvo que sea un usuario entrenado previamente, se puede identificar la autenticidad con acierto.

### 2.2.2. Métodos de adquisición

En este apartado se explica las diferentes formas de obtener los datos de una firma. Una firma es el resultado del movimiento de una mano y un bolígrafo sobre una superficie, tradicionalmente, papel. Existen dos formas de conseguir los datos que resultan de la firma.

#### Adquisición Off-line

Los datos que tenemos en el modo off-line son los extraídos de una firma que ya esta realizada. Siendo más exactos, tenemos una impresión de la misma. Las coordenadas por las que el bolígrafo a pasado. Hay que añadir que los grafólofos son capaces de obtener más datos de una firma off-line. Observando el trazo pueden deducir la presión y velocidad.

En la figura 2.11 se muestran dos formas de adquirir los datos offline de una firma. Se puede realizar mediante una fotografía o mediante escaner.

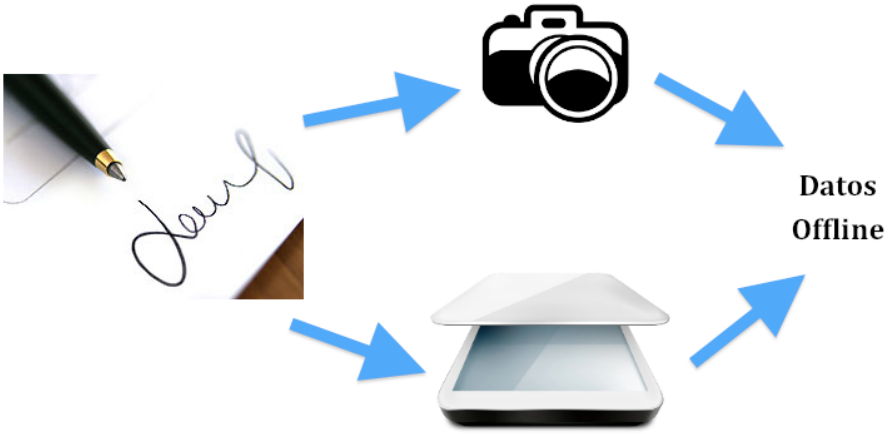



Figura 2.11: métodos de adquisición firma Off-line

### Adquisición On-line

En el modo on-line, los datos son recogidos en el momento de hacer la firma. Se maximiza esta adquisición de datos para un análisis más intensivo de la realización de la firma. Los datos adquiridos permiten saber tanto el movimiento de la punta del boligrafo, la presion y el posicionamiento del boligrafo en el espacio.

- Coordenada “X”
- Coordenada “Y”
- Presion “P”
- Ángulo “Azimut”
- Ángulo “Altitud”
- Tiempo empleado ”Número de puntos”

El dispositivo donde se hace esta adquisición de datos es un aparato con una superficie táctil, en nuestro caso, una tableta gráfica diseñada en principio, para dibujantes. En el mercado hay periféricos especialemente diseñados para la firma como el que podemos ver en la figura.

Figura 2.12: Tableta de la marca Wacom para adquirir la firma On-line(Fuente: wacom.eu)

### 2.2.3. Tratamiento y elección de los datos

La adquisicióin de datos ya ha sido realizada con la tableta pero antes de analizarlos, hay que realizar un ajuste de los mismos, ya que hay factores que varían de una firma a otra.

#### Caracteristicas significativas

Debido a estudios anteriores, en este caso la tesis doctoral "Uso de la Firma Manuscrita Dinámica para el Reconocimiento Biométrico de Personas en Escenarios Prácticos" [9], se sabe que los datos más influyentes en el reconocimiento de firmas son:

- Coordenada "Y"
- Velocidad "V(Y)"
- Velocidad "V(X)"
- Presión "P"

En el sistema creado estas son las características que se tienen en cuenta. Para ello el programa cuenta con un preprocesado que transforma los datos "crudos" que se reciben de la tableta a estos cuatros datos nombrados anteriormente.

#### Normalización de las características

Hay factores que cambian en la realización de la firma. El usuario tiene la libertad de firmar en cualquier parte táctil de la tableta, teniendo así resultados de firmas trasladadas espacialmente. Para solventar este problema se utiliza la normalización geométrica.

- **normalización geométrica** Se utiliza para trasladar los puntos de las firmas siguiendo una condición. A continuación se explican dos formas de realizarla.

  - *Igualación del punto inicial:* sitúa el origen del sistema de coordenadas en el punto inicial de la firma.

$$\begin{aligned} x_n^N &= x_n - x_1 \\ y_n^N &= y_n - y_1 \end{aligned} \tag{2.1}$$

  - *Alineación del centro de masas:* sitúa el origen del sistema de coordenadas(x,y) en el centro de masas de la firma.

$$\begin{aligned} x_n^N &= x_n - \mu_x \\ y_n^N &= y_n - \mu_y \end{aligned} \tag{2.2}$$

- **Normalización estadística:** es la modificación de características para que tengan media cero y varianza uno. Con ella se pretende que su ponderación relativa dentro del vector de características sea la misma [9].

  En la ecuación 2.3, $k$ significa característica, $k$=*[y,p,v(x),v(y)]* y $N$, número de puntos de la firma.

$$\begin{aligned} f_k^N &= (f_k - \mu_k)/\sigma_k \\ \mu_k &= (\sum_{i=1}^{N} f_{k,i})/N \\ \sigma_k &= \sqrt{(\sum_{i=1}^{N}(f_{k,i} - \mu_k)^2)/(N-1)} \end{aligned} \tag{2.3}$$

Hay otros factores externos a la realización de la firma que hay que tener en cuenta. Las características de la tableta y la base de datos que utilicemos también juegan un papel importante en el preprocesado. En el caso que el muestreo sea diferente habrá que adaptarlo para el correcto funcionamiento del sistema. Otros factores como la sensibilidad de las coordenadas X e Y o la presión también hay que comprobarlas y transformarlas si fuera necesario.

### 2.2.4. Aspectos legales y marco tecnológico

Los datos biométricos son considerados de carácter personal por lo tanto su tratamiento está regulado de cara a perservar la privacidad de los usuarios [16]. la Ley Orgánica de Protección de Datos, LOPD, es la que rige como deben ser tratados.

Los requisitos que hay que tener en cuenta cuando se tratan condatos biométricos son los siguientes:

- **El tratamiento de datos ha de ser legítimo**: En éste sentido, se considera legítimo el tratamiento de la huella dactilar de trabajadores a los efectos de llevar a cabo actividades de control horario. Por el contrario, no se considera pertinente el tratamiento de la huella dactilar de clientes para la prestación de servicios comerciales o de alumnos con vistas a controlar su asistencia a un centro escolar.
- **Necesidad del consentimiento del usuario**: Hay que informar y tener autorización del usuario para la toma de datos biométricos.
- **Registro de incidencias**: En caso de problema con los ficheros de la base de datos hay que registrarlo.
- **Control de acceso**: Identificación y autenticación obligatoria para acceder a los datos.
- **Copias de respaldo y recuperación**.
- **Confidencialidad cuando viaja la información**: es necesario cifrar los datos si se mueven por la red.

El artículo 2 de la directiva 95/46/CE del Parlamento Europeo y del Consejo,de 23 de noviembre de 1995 define los datos personales como: "toda información sobre una persona física identificada o identificable (el 'interesado'); se considerará identificable toda persona cuya identidad pueda determinarse, directa o indirectamente, en particular mediante un número de identificación o uno o varios elementos específicos, característicos de su identidad física, fisiológica, psíquica, económica, cultural o social".

Resumiendo, al diseñar un sistema que trata con este tipo de datos hay que tener en cuenta las regulaciones pertinentes anteriormente nombradas.

Sobre el marco tecnológico se puede decir que las herramientas necesarias para su implantación ya están desarolladas en la mayoría de sistemas biométricos (el reconocimiento por olor esta un poco menos desarollado). En la sociedad actual, los sistemas de verificación son los más comunes (hay muchos terminales móviles y ordenadores con lector de huella digital). También se están instalando estos sistemas en lugares como los aeropuertos (el de Madrid-Barajas tiene una verificación en fase de pruebas con lectores de huella).

## 2.3. Algoritmo DTW, Dynamic Time Warping

Una vez descritos los datos de la firma que se van a evaluar, corresponde explicar el algoritmo utilizado en este proyecto.

Hay un gran abanico de posibilidades en cuanto a algoritmos se refiere. Podemos dividirlos en tres clases: métodos basados en fronteras de decisión, modelos estadísticos y por último y los que más nos interesan; métodos basados en alineamiento de características.

Los métdodos basados en alineamiento de características consisten en la comparación de una muestra tomada con una muestra almacenada previamente en la base de datos. El más importante en este campo es el DTW (Dynamic Time Warping) o Alineamiento Temporal Dinámico.

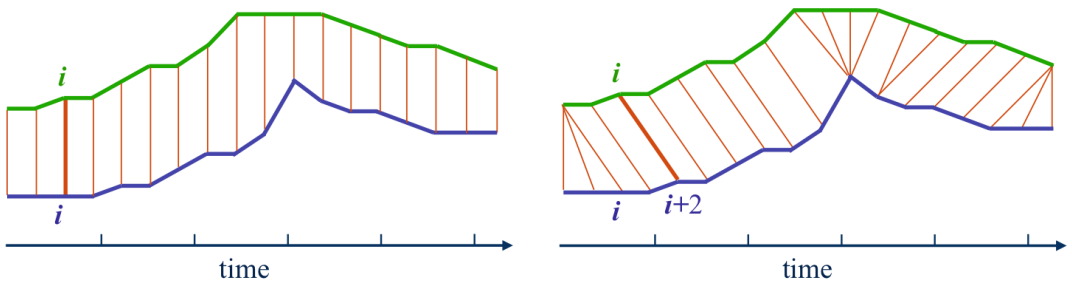



Figura 2.13: izq: sin ejecutar el DTW. drch: tras ejecutar el alineamiento (Fuente: code.google.com/p/gesture-speech-kinect/wiki/Gesture)

El algoritmo se comenzó a utilizar en el reconocimiento de voz y posteriormente se trasladó a otros campos que procesan diferentes realizaciones de la misma

acción en la variable del tiempo, la firma entre ellos. La razón para su utilización es que tiene como gran virtud su capacidad para reconocer similitudes entre dos series aunque se encuentren desplazadas o deformadas [17].

El funcionamiento del mismo se basa en encontrar la distancia más corta entre los puntos de dos series de datos. Se puede decir que este algoritmo hace que una serie sea igual que la otra y el resultado es el coste que esto conlleva. Este alineamiento es no lineal (elástico).

Si introducimos dos series de datos iguales a este algoritmo, el coste será cero. Cuantas más diferencias haya, el coste es mayor. En la figura 2.13 se puede apreciar el funcionamiento a grandes rasgos del DTW.

El DTW se utiliza para procesar secuencias temporales como reconocimiento de voz y gestos, correspondencia de caligrafía, alineamiento de las proteínas y ingeniería química, música y procesamiento de señal[18]. En este caso lo utilizaremos para el reconocimiento de firmas.

### 2.3.1. Desarrollo del algoritmo

Como ya se ha dicho anteriormente, el algoritmo DTW permite realizar un alineamiento óptimo entre dos series vectores con la misma dimensión.

Dadas dos series de vectores A,B de dimensiones D y de longitud "I, J" respectivamente, comenzamos calculando las distancias entre todos sus vectores o muestras. Hay que resaltar que este algoritmo no obliga a que las dos series tengan el mismo número de muestras. Como he descrito anteriormente, la serie A tiene longitud "I" y la serie B, "J".

Las distancias o también llamados costes locales se guardan en la matriz de costes C(IxJ). Cada elemento de la matriz se calcula como muestro a continuación:

$$C_{ij} = dist(a_i, b_j) \tag{2.4}$$

Se pueden utilizar diferentes funciones de costes locales, pero una de las más utilizadas es la distancia euclídea, que es la norma-p de exponente 2.

$$C_{ij} = \sqrt[n]{(a_{i(1)} - b_{j(1)})^n + (a_{i2} - b_{j2})^n + ...(a_{id} - b_{jd})^n} \tag{2.5}$$

Para optimizar el algoritmo se pueden aplicar restricciones para el camino de alineación del dtw. Esta resctriccion o también llamada ventana, hace que no tengamos que calcular todos los costes de la matriz C, sino sólo los que se encuentran dentro de la ventana. El tamaño y la forma de ventana pueden variar. Los más comunes son: La banda de Sakoe-Chiba y el paralelogramo de Itakura.

En la figura 2.14 se muestran dos tipo de restricciones o ventanas. En ambas gráficas estan colocados en el eje horizontal los $N$ puntos de la firma 1 y en el eje vertical los $N$ puntos de la firma 2. En negro se marca el camino DTW (camino con el menor coste local acumulado). También se distinge en gris las dos tipos de ventanas, banda de Sakoe Chiba a la izquierda y paralelogramo de Itakura a la derecha. Se puede ver una clara disminución de coste computacional ya que sólo hay que calcular los costes locales de los valores en gris.

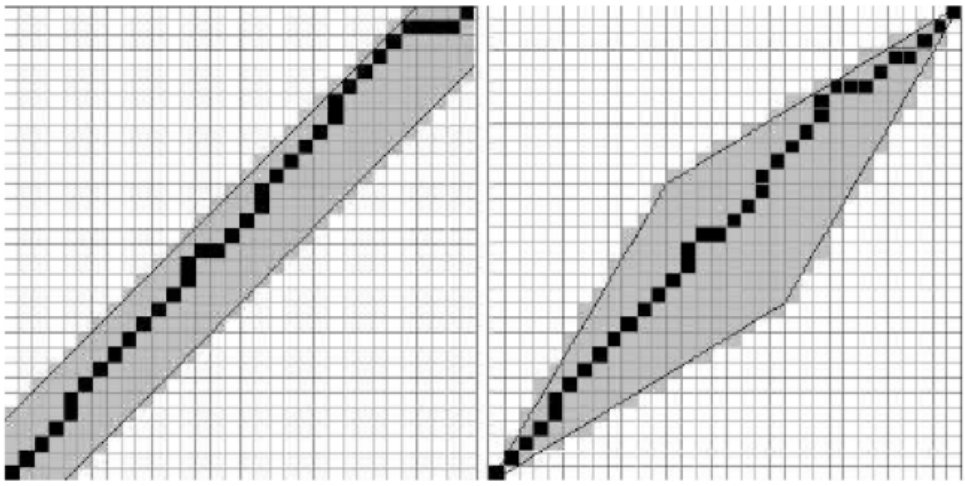


Figura 2.14: izq: Banda de Sakoe Chiba, drch: Paralelogramo de Itakura[10]

Una vez resueltos los valores de coste local que se encuentran dentro de la ventana, hay que proceder a calcular el camino más corto entre el punto C(0,0),primer punto de las firmas y el punto C(I,J) último punto de las firmas.En la figura 2.15 se puede ver la progresión del camino con menos coste. Para este cálculo se hace uso de la restricción de pendiente.

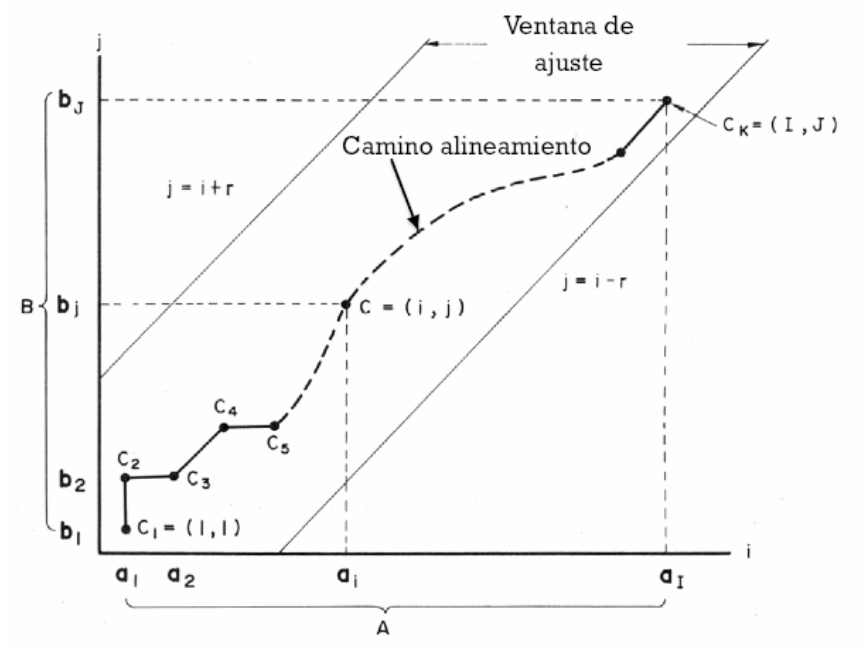



Figura 2.15: Progresión del alineamiento entre 2 firmas (A y B)[4]

La restricción de pendiente usada para alinear 2 conjuntos de datos tiene que cumplir una serie de condiciones. A continuación se muestran estas 5 condiciones. En la explicación la $k$ es el número de puntos que tiene el camino de alineamiento.

1. Monotonicidad: El camino debe tener el sentido de izquierda a derecha y de

abajo a arriba.

$$i_k \leq i_{k+1} \quad j_k \leq j_{k+1} \tag{2.6}$$

2. Continuidad: El paso de un punto de la función al siguiente debe hacerse sin saltos

$$i_k - i_{k+1} \leq 1 \quad j_k - j_{k+1} \leq 1 \tag{2.7}$$

3. Frontera: El punto inicial y el final deben coincidir con el inicio y fin de la matriz.

$$i_0 = 0 \quad j_0 = 0 \quad i_K = I \quad j_K = J \tag{2.8}$$

4. Ventana: El camino no puede salirse de la ventana definida (limite global).

$$\mid i_k - j_k \mid \leq r \tag{2.9}$$

5. Pendiente: se aplican restricciones (límites locales) para reducir la carga computacional del algoritmo.

Llegados a este punto se necesita una segunda matriz, llamada DTW, donde se guardan los valores de la función de alineamiento.Los valores se calculan como se muestra en la siguiente ecuación.

$$DTW[i,j] = \overbrace{C_{ij}}^{\text{coste local}} + \overbrace{min \begin{cases} DTW[i-1,j] \\ DTW[i,j-1] \\ DTW[i-1,j-1] \end{cases}}^{\text{coste acumulado}}$$

El cálculo del coste acumulado puede variar en función de los límites locales que utilizemos. Estos límites locales se rigen por todas los puntos nombrados anteriormente que la función de alineamiento tiene que cumplir. En la figura 2.16 se muestran unos ejemplos de los límites locales, también llamadas restricciones de pendiente:

| P | Esquema gráfico | Simétrico/ Asimétrico | Restricción de pendiente |
|---|---|---|---|
| 0 |  | Simétrico | $\min \begin{bmatrix} DTW(i,j-1)+C(i,j) \\ DTW(i-1,j-1)+2C(i,j) \\ DTW(i-1,j)+C(i,j) \end{bmatrix}$ |
| | | Asimétrico | $\min \begin{bmatrix} DTW(i,j-1) \\ DTW(i-1,j-1)+C(i,j) \\ DTW(i-1,j)+C(i,j) \end{bmatrix}$ |
| 1 |  | Simétrico | $\min \begin{bmatrix} DTW(i-1,j-2)+2C(i,j-1)+C(i,j) \\ DTW(i-1,j-1)+2C(i,j) \\ DTW(i-2,j-1)+2C(i-1,j)+C(i,j) \end{bmatrix}$ |
| | | Asimétrico | $\min \begin{bmatrix} DTW(i-1,j-2)+(C(i,j-1)+C(i,j))/2 \\ DTW(i-1,j-1)+C(i,j) \\ DTW+C(i-1,j) \end{bmatrix}$ |

Figura 2.16: Restricciones de pendiente en el algoritmo DTW[4]

### 2.3.2. Otros algoritmos

- **Métodos basados en fronteras de decisión**: consisten principalmente en la creación de fronteras de decisión entre clases en función de un criterio de error. Este criterio se crea a partir de la relación entre los resultados obtenidos y los deseados.

  Como ejemplos de estos métodos podemos encontrar las redes neuronales, los árboles de decisión y las máquinas de vectores de soporte.

- **Métodos basados en modelos estadísticos**: estos métodos se basan en el uso de patrones de referencia para elaborar modelos probabilísticos y/o estadísticos.

  Los métodos más utilizados de este área son el algoritmo HMM (Modelos Ocultos de Markov) y el GMM (Modelo de Mezcla de Gaussianas).

## 2.4. Raspberry Pi

En este apartado se explican todo lo relacionado con el hardware relativo a este proyecto. Se explicarán tanto las características del procesador elegido, como otras opciones que se tuvieron que descartar. Finalmente se ha utilizado el Raspberry Pi 1 Model B+ por su bajo precio, alta conectividad, y grandes posibilidades de expansión en el futuro.

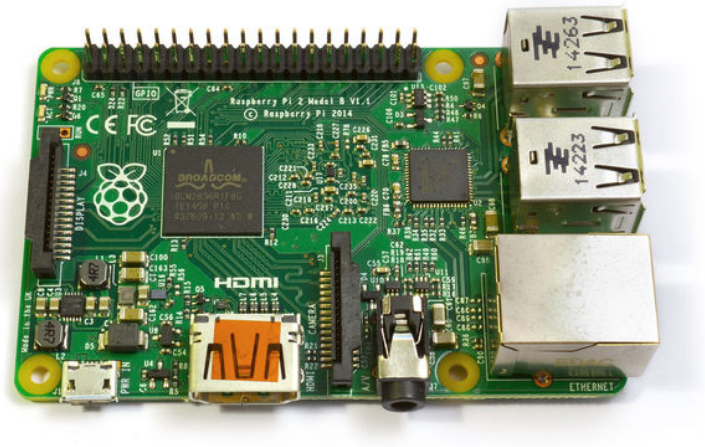



Figura 2.17: Raspberry Pi 1 model B+ (Fuente: networkworld.com)

### 2.4.1. Características

El cerebro de este sistema es un procesador con arquitectura ARM, muy utilizado hoy en día en teléfonos móviles por su buena relación rendimiento/energía consumida.

La figura 2.18 muestra la sencilla arquitectura de este modelo de Raspberry Pi. De izquierda a derecha, bloque de GPIO (Gerneral-Purpose Input/Output), RAM

(Random Access Memory), CPU(Central Process Unit),GPU(Graphic Process Unit) y USB hub.

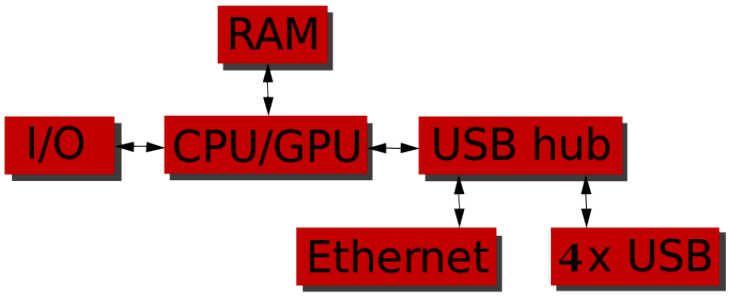



Figura 2.18: Arquitectura de Raspberry Pi 1 model B+ (Fuente: wikipedia.com)

A continuación, se explican las características de este hardware:

- Procesador: "ARM1176JZF-S" de la familia ARM11 que tiene un reloj de 700MHz y una cache L1 de 16KB y L2 de 128KB. Este modelo sólo tiene un núcleo.
- Memoria: Cuenta con 512 MB de memoria SDRAM ( Synchronous Dynamic Random Acces Memory) a una frecuencia de 400 MHz. Esta memoria es compartida entre el procesador y la gráfica y el tamaño se asigna dinamicamente.
- Gráfica: cuenta con un coprocesador Broadcom VideoCore IV a 250 MHz. Gracias a su conexión HDMI (Hight Definition Multimedia Interface) soporta resoluciones de hasta 1920x1200 (WUXGA). Cabe destacar que este procesador es capaz de reproducir video a 1080p.
- Almacenamiento: Este procesador empotrado cuenta con una ranura microSD que soporta memorias de hasta 32 GB. El sistema operativo esta instalado en una tarjeta de 16 GB.
- Periféricos: Tiene 17x GPIO, 1x salida de Audio (3.5mm jack), 4x USB 2.0, 1x 10/100Mbit/s Ethernet (RJ45), 1x HDMI 1.4, 1x microSD slot (hasta 32 GB).
- Sistemas Operativos: existen múltiples sistemas operativos modificados para este dispositivo, la mayoria son Unix como Raspbian (un Debian modificado para Raspberry Pi), Ubuntu o Archlinux. Windows 10 ha hecho una extensión para utilizar este dispositivo en el IOT (internet de las cosas).

### 2.4.2. Otras opciones

Dentro de este tipo de hardware hay múltiples opciones en el mercado. Como veremos a continuación hay pequeñas variaciones entre los mismos, pudiendo elegir el que más se adapte a las funcionalidades y rendimiento que se están buscando. Entre las más significativas se pueden citar:

- **Xilinx Zynq-7000:** además llevar el chip propio de una FPGA (Field Programmable Gate Array) en este caso un Artix-7, con 85K puertas lógicas, cuenta con un procesador dual-core ARM Cortex-9 y 1GB de DDR3. Tanto esta placa como la explicada a continuación (Nvidia Jetson) son muy superiores en comparación con las que se van a detallar posteriormente, tanto en potencia de procesamiento como en precio.

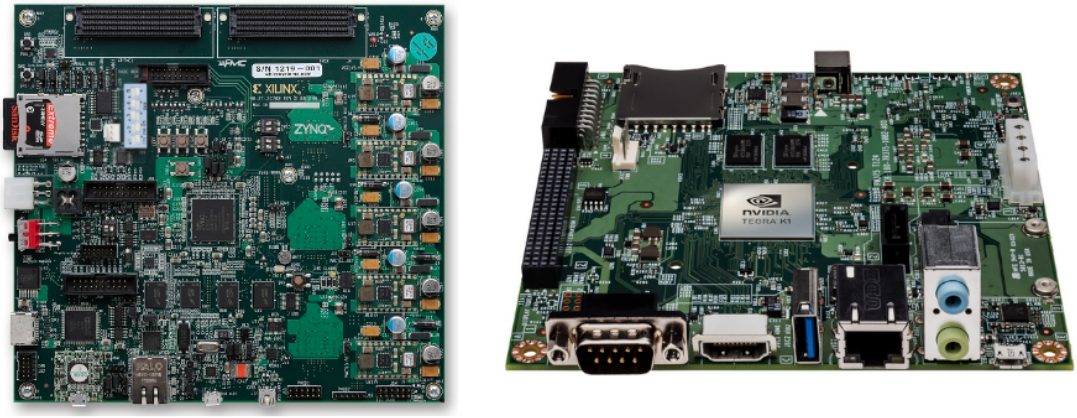



Figura 2.19: izq:Xilins Zynq-7000; drch: Nvidia Jetson TK1 (Fuente: xilinx.com y nvidia.com).

- **Nvidia Jetson TK1**: equipado con un ARM quad-core Cortex A-15 y un GPU Keppler con 192 núcleos CUDA (Compute Unified Device Architecture), 2 GB de memoria RAM y 16 de memoria eMMC, hace que tenga mucho potencial si sabes sacarle partido a los núcleos CUDA por su gran capacidad de paralelización.

- **Intel Edison:** cuenta con una potencia de un procesador doble núcleo Intel Atom a 500MHz y un microcontrolador mono núcleo destinado a la gran recogida de información de los 40 peurtos GPIO. Cuenta con 1 GB de DDR (Double Data Rate) y 4 GB de memoria Flash.Tiene alta conectividad (Wifi y Bluetooth).Esta diseñado para su aplicacion en campos como Ïnternet of Things"(IoT) o la ropa inteligente (Wereable Technology).Corresponde con el dispositivo colocaldo a la derecha en la figura 2.20.

- **BeagleBone (Texas Instruments):** con un procesador ARM Cortex -A8 a 1GHz y una memoria DDR de 512 MB es muy similar al Raspberry Pi 2, el nuevo modelo de la compañia, ya que además cuenta con un puerto USB de salida y un puerto de red. Cabe destacar que se pueden añadir extensiones a este dispositivo para aumentar sus funcionalidades como controladores VGA, LCD,controlador de motores, baterías o una placa protoboard. Se puede ver en la parte izquierda de la figura 2.20.

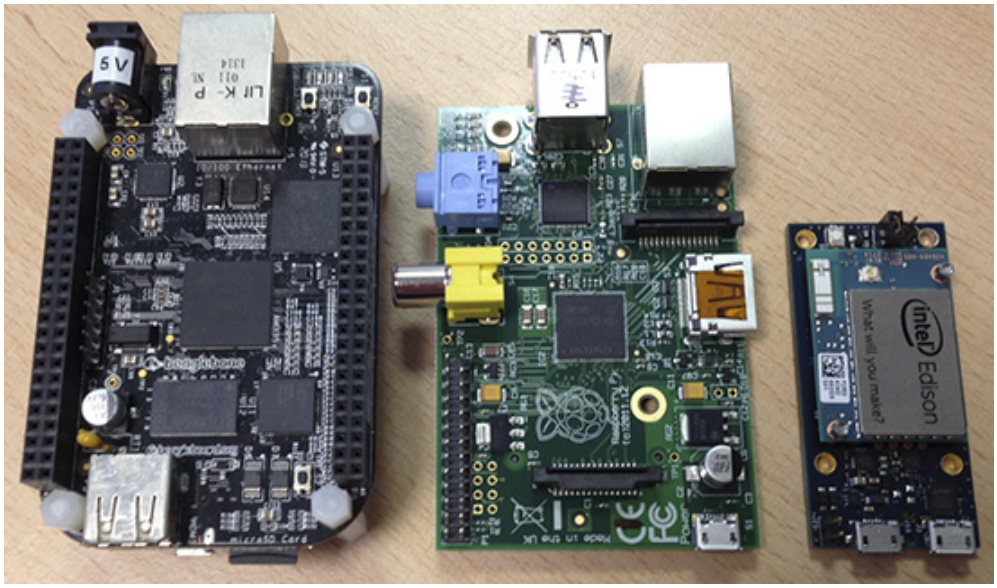



Figura 2.20: izq: BeagleBone; centro: Raspberry pi 1(Modelo A); drch: Intel Edison con mini placa de desarrollo (Fuente: davidhunt.ie).

- **MicroBlaze:** Es un "soft processor", un procesador implementable sobre lógica programable. Este modelo esta diseñado para las placas Xilinx. En su configuración se puede elegir desde el tamaño de la memoria hasta el número de núcleos. Este procesador puede funcionar con un kernel de Linux, lo que junto con toda la lógica de la FPGA que no se ha utilizado para crear el microprocesador puede ser una opción muy potente en términos de computación.

- **Arduino y Núcleo ST (STM Microelectronics):** se han estudiado estas dos opciones pero debido a su limitación de procesador y de memoria, habría que añadir alguna extensión para almacenar la base de datos.

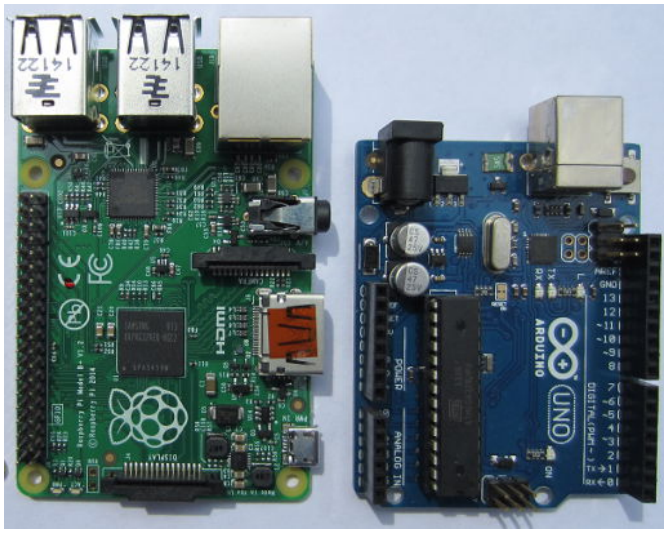



Figura 2.21: izq: Raspberry pi 1(Modelo B); drch: Arduino One(Fuente: reuk.co.uk).

### 2.4.3. Elección del hardware

Finalmente el proyecto se ha realizado en una Raspberry Pi debido a su alta conectividad, a la posibilidad de tener un sistema operativo Unix, Raspbian en este caso. También por su bajo precio, 35 Dólares.

## 2.5. Comunicación por USB

Para que el sistema funcione se necesita leer los datos de la tableta Wacom. La conexión se realiza por medio de un puerto USB (Universal Serial Bus). Para adquirir los datos se ha realizado un estudio sobre el puerto USB y un driver, a continuación se explican los conceptos necesarios para el correcto entendimiento del mismo.

### 2.5.1. Introducción

El puerto USB fue diseñado en 1996 y fue fruto del trabajo de USB-IF( USB Implementers Forum) asociación en la cual estaban presentes las empresas: Intel, Microsoft, Compaq, LSI, Apple y Hewlett-Packard. Este puerto se diseñó para sustituir a los puertos serial (tanto el RS232 como el PS/2) y paralelo utilizados para conectar periféricos.

Una de las principales caracteísticas es el poder conectar los periféricos "en caliente", característica que los anteriores puertos no tenían. Había que apagar el sistema para conectar un periférico. Actualmente cuenta ya con cuatro versiones importantes:

- versión 1: Esta versión contaba con un modo de "Baja velocidad" a 1.5 Mbit/s y "Alta velocidad" a 12Mbit/s.
- versión 2: publicada en el año 2000, cuenta con una tasa máxima de 480Mbit/s. Es el puerto estándar a día de hoy.
- versión 3.0: anunciada en el año 2008, tiene una tasa de transmisión de 5Gbit/s, 10 veces superior a la anterior versión. Desde hace al menos 2 años es fácil encontrarlo en los ordenadores de gama media-alta.
- versión 3.1: la ultima, presentada en 2013 garantiza una tasa de 10 Gbit/s. Esta versión del puerto es muy especial ya que admite mucha transferencia de energía, incluso para poder alimentar ordenadores portátitles. Actualmente hay muy pocos modelos que lo tengan integrado, siendo el nuevo Macbook uno a destacar ya que sólo lleva ese puerto para tanto cargar o conectar cualquier dispositivo.


Figura 2.22: USB 3.1 en Macbook (Fuente: Apple.com).

### 2.5.2. Transmisión de datos

El USB tiene tanto conectores propios como protocolos de comunicación. Cada periférico tiene un maximo de 16 direcciones de memoria de entrada y 16 de salida[14]. Estas direcciones o "endpoints" son los puntos de lectura/escritura Hay cuatro tipos de interacción con el periférico para transmitir información (los cuatro son bidireccionales):

- Control: Tiene el mismo "endpoint" de entrada y de salida. Sirve para la configuración inicial de dispositivo.
- Isocrona (Isochronous en ingles): Tiene un ancho de banda garantizado pero la información puede contener errores ya que no se comprueba la integridad de los datos. Aparatos como la webcam o el micrófono utilizan este tipo de transmisión.
- Interrupcion: libre de errores y baja tasa de transmisión. Los dispositivos que utilizan esta transmisión son los ratones y teclados, pero también otros como las tabletas Wacom.
- Bulk: Esta transmisión se utiliza para mover gran cantidad de datos. No garantiza el ancho de banda, pero si la integridad de los datos. La principal utilidad de este tipo es mover datos a memorias externas.

Todas los tipos de transferencias leen o escriben los paquetes en un "endpoint". En el caso que es más interesante para estre proyecto, las transferencia por interrupción, el endpoint lo marca el periférico.

### 2.5.3. Controlador

Tras exponer las características generales de la conexión USB, este apartado se va a centrar en los aspectos que harán posible la lectura de los datos de la Wacom.

Dado que sólo se necesitan datos de entrada de un dispositivo con leer un "endpoint" es suficiente. Como punto de partida sólo se sabe la dirección de memoria donde hay que leer la informació (facilmente visible en propiedades del usb), pero no se sabe ni el orden, ni el formato ni la longitud de los datos. Todos estos valores se han solucionado estudiando los drivers oficiales de Wacom y posteriormente implementando un pequeño programa de lectura del "endpoint".

Como ya se explicará posteriormente, este método funciona en los ordenadores habituales con arquitectura X86 pero no en el microprocesador empotrado con arquitectura ARM donde este proyecto se ha desarrollado aunque tenga instalado un kernel de Linux (Debian).

## 2.6. Base de datos MCyT

La base de datos utilizada para el ajuste del algorítmo es la MCyT. Esta base de datos no sólo contiene firmas sino también huellas dactilares. Dado que este trabajo se centra en la firma, vamos a detallar todo lo referente a ella.

Varias universidades son las responsables de la existencia de esta base de datos de 300 usuarios, aquí podemos ver detalladamente la proveniencia de las mismas:

- Universidad Politécnica de Madrid: 145 usuarios
- Universidad de Valladolid: 75 usuarios
- Universidad del País Vasco: 75 usuarios
- Escuela Universitaria Politécnica de Mataró: 35 usuarios

Este conjunto de datos consta de 300 usuarios. Para la creación de esta base de datos, cada usuario crea 25 firmas verdaderas y realiza otras 25 falsas intentando falsificar otros usuarios de la base de datos.

Las firmas fueron captadas con una tableta Wacom con características muy similares a la que se utiliza en este proyecto.

### 2.6.1. Características

- Número de firmas: tenemos un total de 5000 firmas las cuales la mitad son verdaderas y la otra mitad falsificaciones.
- información obtenida de la tableta:
  - Coordenada x.
  - Coordenada y.
  - Presión ejercida.
  - Ángulo Azimut.
  - Ángulo de altitud.
- wacom: la resolución de la tableta es de 100 lineas por mm y la precisión es de 0.25mm(igual que la utilizada en este proyecto) . La frecuencia de muestreo es de 100Hz y la presión tiene un rango de 1024 niveles. Ambos ángulos tienen una resolución de 36 segundos (la centésima parte de un grado).

### 2.6.2. Otras bases de datos

- Philips[13]: Cuenta con 51 usuarios, 30 firmas verdaderas y hasta 70 falsas por usuario. Se puede considerar que los datos corresponden a firma on-line ya que tenemos las 5 características que he nombrado anteriormente.
- Biomet: esta base de datos contiene 84 usuarios con 15 firmas genuinas y hasta 12 falsas por usuario. Son firmas on-line.
- SUSIG Blind Subcorpus: cuenta con 100 usuarios, 8 firmas verdaderas y 10 falsas por usuario. Las características almacenadas son la posición y la presión.
- MyIDea: también con 100 usuarios , 18 firmas lícitas y 18 ilicitas por usuario. Cuenta con los 5 parámetros en su caracterización.
- BuoSecurlD : Es la más extensa en cuanto a usuarios y cuenta con 400. Cada uno de ellos tiene 16 firmas verdaderas y 16 falsas. Los 5 parámetros fueron adquiridos en esta base de datos.

### 2.6.3. Eleción de MCyT

Finalemente se ha elegido la base de datos MCyT por su gran cantidad de datos (25 firmas verdaderas y 25 falsas por usuario), por su alta resolución en las características de la misma y por la posibilidad de utilizar una parte de la misma (100 usuarios) gratuitamente.

Otra característica de esta base de datos que hace que sea buena para le proyecto es que el dispositivo con que el fueron tomadas las muestras (Wacom) es muy similar al que este proyecto utiliza.

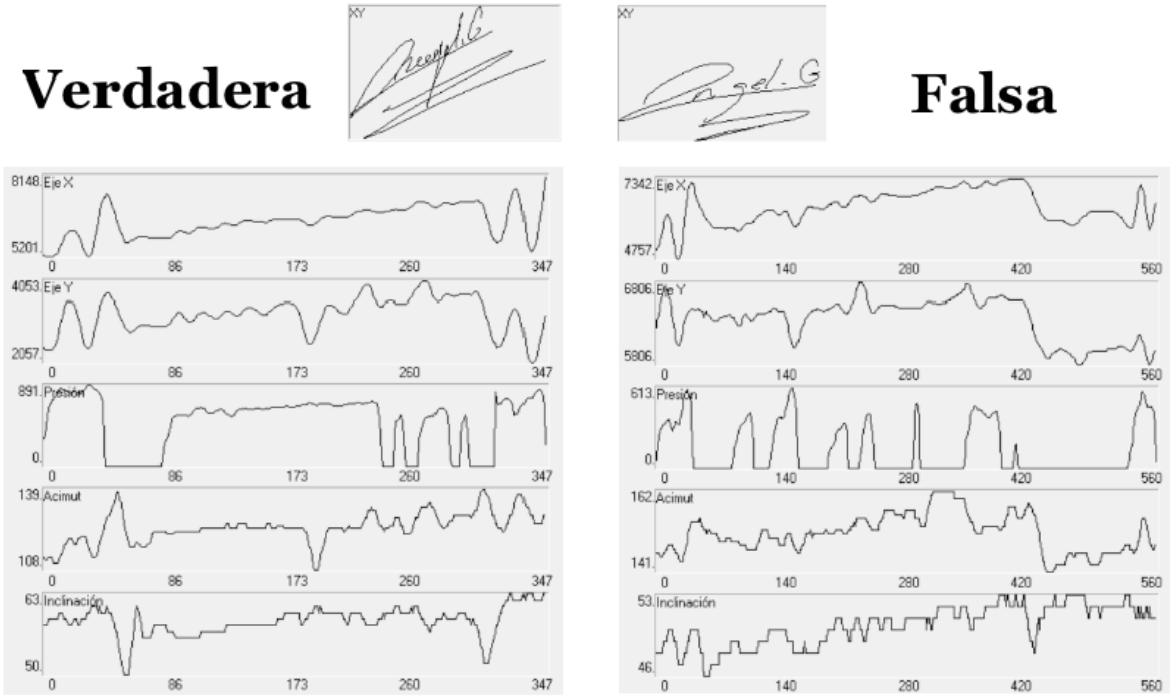



Figura 2.23: Comparación de firmas de la base de datos MCyT (usuario 37)

# Capítulo 3

# Desarrollo del sistema

Una vez se han explicado los conocimientos previos para el correcto entendimiento de este proyecto, se da paso a este capítulo donde se desglosa el sistema construido. Se expondrá tanto los pasos que se han dado, elecciónes que se han tomado como el funcionamiento del resultado final, incluyendo los contratiempos.

## 3.1. Elecciónes de diseño

En esta sección se explican las decisiones que se han tomado antes y durante la realización del proyecto. Estas decisiones engloban tanto hardware como software. Son las que definen a grandes rasgos el resultado final obtenido.

### 3.1.1. Hardware

La primera elección fue que hardware se va a utilizar. En primer lugar, la tableta donde se iba a realizar la firma. Tras buscar diferentes opciones en mi entorno y en el mercado, se eligió una Intuos 4 PTK 440 por su gran versatilidad, sus buenas especificaciones, su amplia area sensitiva y su fiabilidad, (este modelo lleva 5 años en el mercado).

Dado que la conexión de la tableta es USB, se buscó un hardware que tuviera esta conexón.

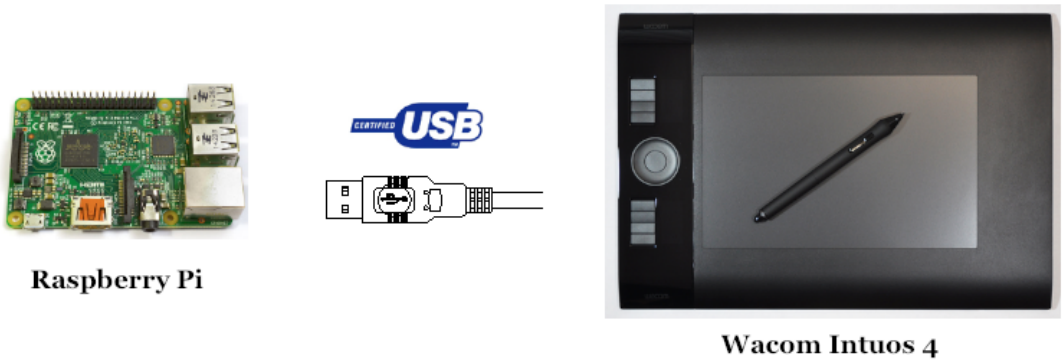



Figura 3.1: conexión entre Raspberry Pi y Wacom

De las distintas opciones analizadas en un apartado 2.4.2, se eligió el Raspberry Pi por su alta conectividad, su bajo precio, y su posibilidad de instalar algún tipo de linux. Así la comunicación Wacom-procesador se presenta más sencilla que otras opciones, como por ejemplo la utilización de un conversor usb-serial para conectarlo la un FPGA o pagar por un driver privativo (para utilizar el puerto usb de una FPGA).

### 3.1.2. Lenguaje programación

Dado que el sistema tiene que tener una buen rendimiento para que los tiempos de espera para el usuario sean mínimos se ha elegido el lenguaje C. Otra razón para esta elección ha sido el previo conocimiento del lenguaje por parte del alumno y la amplia documentación y bibliotecas que existen.

### 3.1.3. Base de datos

El sistema esta diseñado para que dada una firma de entrada, busque el usuario al que pertenece en la base da datos. Toda esta información se almacena dentro de un subdirectorio "db_firmas" localizado en el directorio donde esta el ejecutable.

El numero de firmas guardadas para el funcionamiento del sistema son 5 verdaderas por usuario. Con este conjunto se solventarán los problemas de variabilidad en las realizaciones de las firmas. Este numero de firmas es suficiente para conseguir unos buenos resultados [6].

Para que el sistema sea robusto, se crea un archivo por cada firma que tiene que almacenar en la base de datos, de esta forma, en caso de que un archivo se dañara, el programa sigue funcionando correctamente, aunque la capacidad de detección se vería reducida. El nombre del archivo da la información sobre que numero de usuario pertenece, si es verdadera o falsa y el numero de firma.

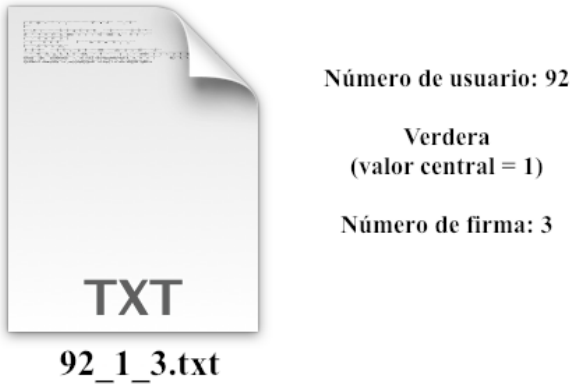



Figura 3.2: Archivo firma de la base de datos del sistema

La razón para tener un valor en el nombre para indicar si la firma es verdadera o falsa reside en la fase de entrenamiento. Durante esta fase se han necesitado firmas falsas de usuarios. En el funcionamiento normal del sistema no es necesario tener ninguna firma falsa en la base de datos.

En cuanto a la seguridad de los datos; aunque no estén cifrados (según la ley no es necesario a menos que viajen por la red), no son de fácil acceso. Los datos

no se pueden leer con un simple editor de texto ya que la información que contiene el fichero es una estructura de C copiada bit a bit.

### 3.1.4. Parámetros del algoritmo DTW

El algoritmo DTW no es un algoritmo estático. Tiene parámetros en cuanto a su implementación que pueden ser variables, como ya se ha explicado anteriormente. Se ha realizado una búsqueda de los mejores parámetros con los que ajustar el sistema. Gracias a un trabajo de fin de grado de esta universidad [5] que estudiaba las mejores funciones de alineamiento para el algoritmo DTW se llegó a la conlusion de que la P(0) asimétrica mostrada a continuación es la mejor para esta implementación. En este estudio se muestran otras funciones de alineamiento que mejoran levemente los resultados finales, pero la diferencia del coste computacional es sustancial.

| P | Esquema gráfico | Simétrico/ Asimétrico | Función de alineamiento |
|---|---|---|---|
| 0 |  | Asimétrico | $\min \begin{bmatrix} DTW(i,j-1) \\ DTW(i-1,j-1)+C(i,j) \\ DTW(i-1,j)+C(i,j) \end{bmatrix}$ |

Figura 3.3: funcion de alineamiento o coste local utilizado en el sistema[4]

En la implementación del DTW también se hace una normaliación sacada del trabajo de Sakoe Dynamic Programming Algorithm Optimization for spoken word recognition [4] respecto al valor de salida de la función DTW. Esta normalización intenta ponderar el valor según la longitud de las firmas.

$$DTW^N = \frac{DTW}{longitud(firma(1)) + longitud(firma(2))} \tag{3.1}$$

Esta ponderación sirve para evitar en lo posible que en la comparación entre parejas de firmas cortas y firmas largas, el resultado del algoritmo no varíe en función del tamaño de la firma sino en el de las diferencias entre ellas.

### 3.1.5. Preprocesado de los datos adquiridos

Aquí se explican las decisiones que se han tomado sobre las transformaciones necesarias de los datos de entrada recogidos con la tableta (o los que provienen de la base de datos MCyT) para su posterior procesado del algoritmo.

Se expoden tanto las normalizaciones que se hacen a las características como la interpolación utilizada.

#### Características elegidas

Los datos de entrada tanto de la base de datos MCyT como en la Wacom, son vectores de caracteísticas que ha leído la tableta. Este vector contiene dos tipos de características, las posicionales, puntos en 2D por donde ha pasado el bolígrafo, y las ergonómicas (presion y ángulos inclinación y azimut).

Hay varias formas de elegir las características que se van a analizar. En este caso se han elegido siguiendo el trabajo [9], donde se estudian que grupo de características son las más significativas para su posterior análisis con el DTW.

En el estudio se busca tanto las mejores características como la mejor convinación entre ellas. Se evalúan desde posición, presión, angulos hasta sus derivadas respecto al tiempo, la velocidad y aceleración. La conclusión a la que el autor llega es que el conjunto de características más significativo es el siguiente:

$$\begin{aligned} S &= y, dy, dx, p \\ dx &= (x_{t+1} - x_t)/\Delta t \\ dy &= (y_{t+1} - y_t)/\Delta t \end{aligned} \tag{3.2}$$

La $y$ es la coordenada Y, $dy$ es la velocidad de la coordenada Y (la primera derivada respecto del tiempo de Y), la $dx$ es la velocidad de la coordenada X, y la $p$ es la presión ejercida

### Normalización geométrica

Debido a la libertad a la hora de firmar en el sensor en cuanto a donde comienza la firma se aplicó una normalización donde se situaba el origen de la firma en el origen de coordenadas, *igualación del punto inicial*.

$$\begin{aligned} y_t^N &= (y_t - y_0) \\ x_t^N &= (x_t - x_0) \end{aligned} \tag{3.3}$$

Posteriormente se implementó una normalización diferente, *igualacion del centro de masas*, que mejoraba ligeramente los resultados. Se situó el origen de coordenadas en el centro geométrico de la firma.

$$\begin{aligned} y_t^N &= (y_t - \mu_y) \\ x_t^N &= (x_t - \mu_x) \\ \mu_k &= (\sum_{i=1}^{N} f_{k,i})/N \\ k &= x, y \end{aligned} \tag{3.4}$$

### Interpolación

En este apartado se explica las diferentes interpolaciones que se han utilizado. Son varias ya que se ha tenido que pasar de una a otra por razones de rendimiento.

Se ha visto por las diferentes ejecuciones de este algoritmo que aunque pueda procesar firmas de diferentes tamaños, los resultados son mejores si todas las firmas tienen el mismo numero de puntos.

Ademas haciendo una interpolación reducimos el muestreo con lo cual el coste computacional es menor en la comparación de las firmas. Este muestreo se nombrará para futuras referencias como **muestreo general**.

- **Interpolación por redondeo**

  En un primer momento se utilizó una interpolación por redondeo. Esta interpolación modifica las firmas que tienen más puntos que el muestreo marcado. Las firmas que tienen menos puntos no son modificadas.

  La figura 3.4, es un gráfico sobre la interpolación por redondeo. En negro están los puntos que corresponden a la firma sin interpolar (captados por por la tableta y normalizados geométricamente). La separación de los puntos negros corresponde al periodo correspondiente al muestreo de la tableta (ecuación 3.6). En azul están los puntos de la interpolación separados temporalmente el periodo de la interpolación, ecuación 3.5.

  Los valores que toman los puntos en azul son los que corresponden al valor más cercano temporalmente de los puntos negros (puntos originales). Por esta razón el valro de t=1,6 se iguala con el valor de t=2 y el valor de 3,2 se iguala con el valor que tiene t=3.

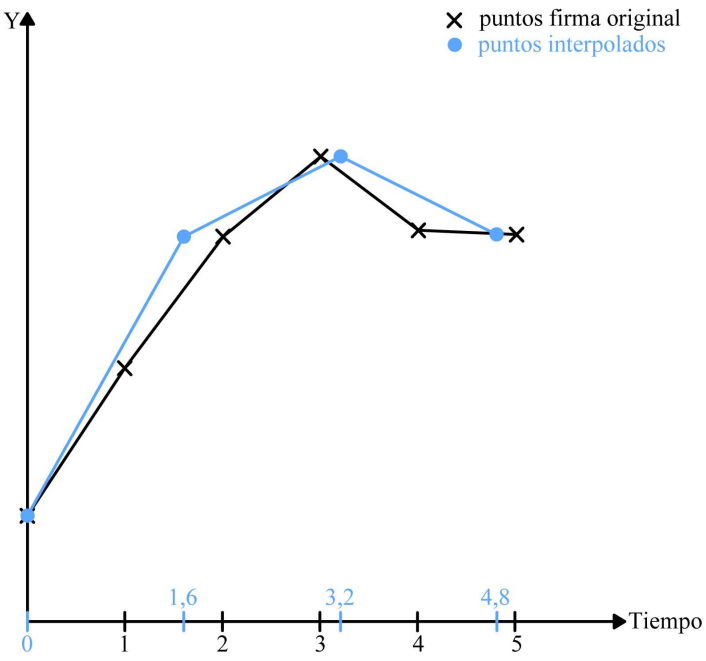



Figura 3.4: Interpolación por redondeo

En la equación 3.5 la variable $N$ es el número total de puntos de la firma original, y *Muestreo general* es el número total de puntos de la firma interpolada.

$$T_{interpolacion} = N/Muestreo\ general \tag{3.5}$$

$$T_{muestreo} = 1/frecuencia\ muestreo \tag{3.6}$$

- **Interpolación lineal**

  Finalmente se ha utilizado una interpolación lineal. Se ha cambiado la interpolación por razones de rendimiento. Con esta interpolación el error se reduce en torno al 0,6 %.

  Las firmas de salida de esta interpolación siempre tienen el mismo número de puntos. Ya que aunque alguna firma tenga menos puntos que el muestreo elegido, la interpolación busca los puntos necesarios según una aproximación lineal.

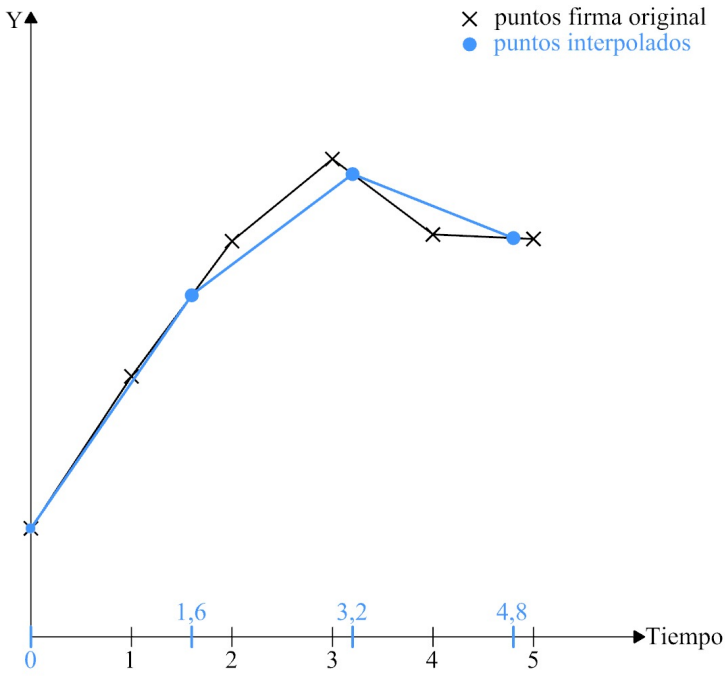



Figura 3.5: Interpolación lineal de la coordenada Y

## 3.2. Desarrollo del software

En este apartado se explica como se ha desarrollado el programa para un correcto funcionamiento con el hardware, siguiendo las elecciones anteriormente nombradas en el apartado 3.1. Se explican con detalle; la lectura de datos de la conexión USB, la estructura utilizada para almacenar las firmas, la forma de implementar el algoritmo DTW y el clasificador utilizado.

La figura 3.6 resume en gran medida este apartado. Este gráfico es un esquema de como fluyen los datos dentro del programa y como se realiza la comunicación con partes externas al programa (tableta y base de datos). Dentro de la figura se puede diferenciar 3 diferentes partes (divididas por lineas verticales amarillas) que el programa necesita para su funcionamiento. De izquierda a derecha son: Perifericos (tableta Wacom conectada por USB),Memoria RAM ( lugar donde se encuentra todo el programa durante su ejecución), y Disco Duro (memoria donde se localizan la base de datos del programa y también la base de datos MCyT).

En esta figura (3.6) se puede apreciar también el diseño modular del mismo. Estos módulos nos da mucha libertad para cambiar funciones y parámetros si fuera necesario.

Para la correcta interpretación del gráfico hay que tener en cuenta que el flujo de datos que comienza en MCyT (en naranja) sirve para ajustar el sistema y añadir usuarios a nuestra base de datos. En el modo de funcionamiento normal, esta parte del flujo esta desactivada ya que se añaden y detectan usuarios desde la tableta Wacom.

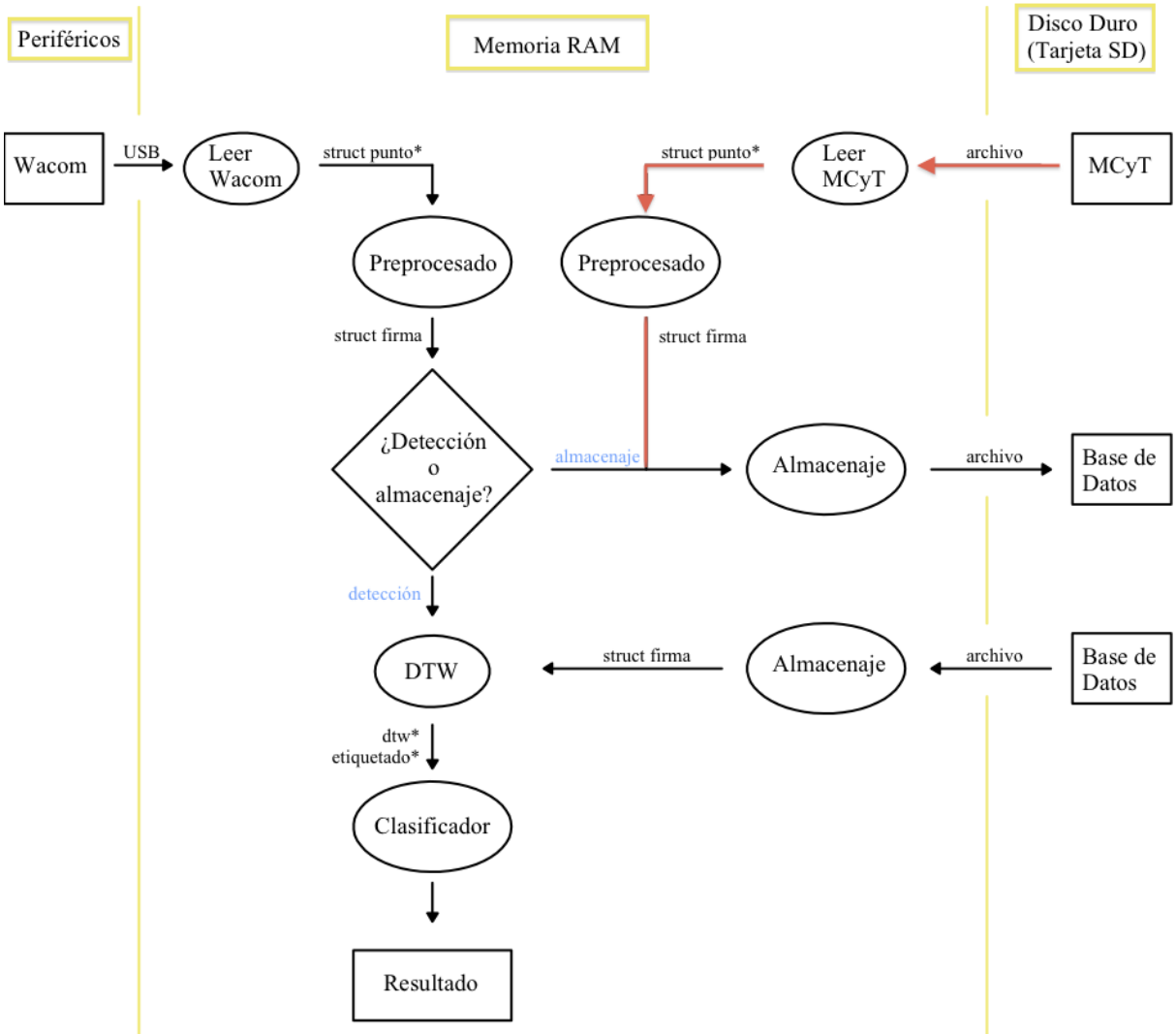



Figura 3.6: Esquema de funcionamiento del sistema.

### 3.2.1. Adquisición de datos

Como se puede ver en la figura 3.6, hay dos formas de adquirir los datos de firmas: desde la base de datos MCyT y desde el lector Wacom. Ambos se han implementado íntegramente para este sistema.

#### Lectura MCyT

Esta base de datos cuenta con una función para Matlab para su lectura. Dado que el programa esta en C, no es válida. Se ha creado una función (leer MCyT) para su correcta lectura siguiendo la estructura de la función en Matlab.

Solo se puede resaltar la necesaria utilización de la biblioteca "stdind" debido a que los datos están en formato uint16_t ( integer de 16 bit) y uint32_t (integer de 32 bits).

#### Lectura Wacom

Dado que el sistema se ejecutaría sobre un kernel de Linux (en particular el 3.12) , el proyecto se comenzó a desaroyar en un ordenador con Ubuntu instalado. Se desarolló un driver que funcionaba correctamente en esta plataforma, pero al probarlo en el procesador embebido, no leía los datos. Este problema se explicará más detalladamente en el apartado de dificultades, 3.3.1.

Finalmente se llegó a la conclusión que la mejor forma de leer los datos de entrada es leer un buffer del hub USB que se puede acceder a él ejecutando la un sistema de archivo de archivos llamado **debugfs**. Este sistema fue diseñado para propositos de depuración. En nuestro caso el modulo que se utilizará es **usbmon**.

En este buffer se pueden leer tanto los datos de salida como los de entrada de todos los usb conectados en el sistema. El campo "tipo de conexión"(variable "T" en la figura 3.8) es el que muestra que tipo de transmisión de datos es la trama USB. Este campo puede ser: Control, Isocrona, Bulk e Interrupción. (Mas información en 2.5.2).

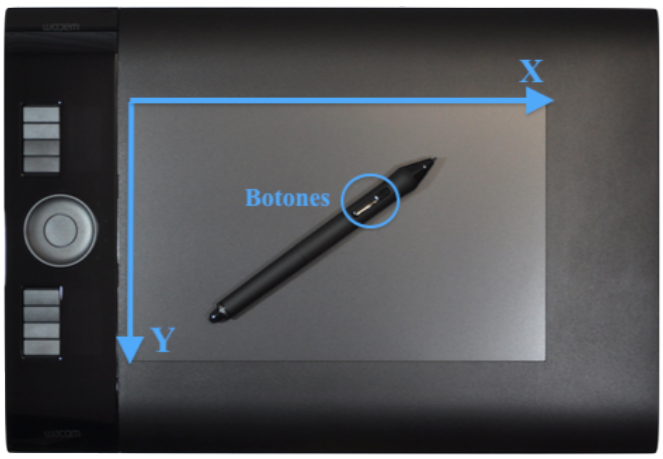



Figura 3.7: Tableta wacom con la orientación de los ejes

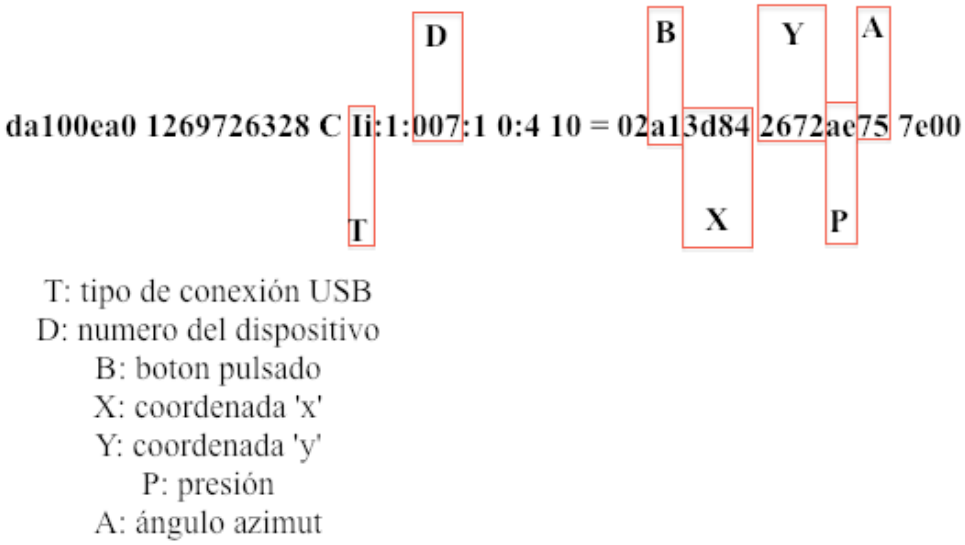



Figura 3.8: Trama de puerto usb leída del buffer

En la figura 3.8 se ve una trama USB adquirida con este método. Se han marcado los diferentes campos importantes para el funcionamiento del programa. La trama que se expone es del tipo 'Ii':Interruption Input, interrupción de entrada [7]. Este el el formato con el que la tableta se comunica. La 'D'corresponde con el número del dispositivo que el sistema asigna a este periférico. Este número cambia cada vez que reiniciamos la tableta por lo que es el único parámetro de entrada en la ejecución del sistema. La 'B'corresponde con la pulsación de los botones situados en el lápiz. La 'X'corresponde con la coordenada horizontal en hexadecimal. La 'Y'corresponde con la coordenada vertical, la 'P'corresponde con la presión. El ángulo azimut corresponde con la 'A'.

Toda esta información se ha deducido haciendo haciendo diferentes pruebas. Para conseguir la posicion X e Y en la trama se ha contrastado la información de las tramas colocando el lápiz en las diferentes esquinas de la tableta. También para conseguir la presión y el ángulo.

Tras las pruebas se llegó a identificar tanto los datos dentro de la trama como la orientación de los ejes en la tableta, visible en la figura 3.7.

Después de identificar los campos necesarios para el buen funcionamiento del DTW, se creó una función de lectura de este buffer. Este métdodo hace las conversiones necesarias para que los datos sean compatibles con el resto de programa.

### 3.2.2. Preprocesado

En el programa se ha implementado el modulo de preprocesado. En este módulo se encarga de normalizar e interpolar los datos. Esta parte solo se ejecuta cuando entra una firma nueva al sistema pues las firmas almacenadas en la base de datos ya están preprocesadas.

En un principio se implementó este bloque con una normalización geométrica por *igualación del punto inicial*, explicada en el apartado 3.1.5. Por motivos de

rendimiento se cambió a la normalizacion por *centro de gravedad*. Se ha hecho la comparación de los 2 tipos de normalización con 2 conjuntos de 500 firmas (100 usuarios con 5 firmas verdaderas y 5 falsas cada uno) y en el peor de los casos se ha mejorado un 2.5 % el error (punto EER).

Una vez realizada la normalización geometrica se realiza la interpolación. El valor con el que se realiza la interpolación se llama *Muestreo general* yse ha ajustado previamente tras hacer pruebas.

La información sale de este módulo en forma de una estructura (struct firma). Se pueden ver todos sus campos en le Figura 3.9.

### 3.2.3. Base de datos

Se ha creado una base de datos para que el sistema almacene las firmas de los usuarios. Gracias a ella es posible detectar un usuario que previamente se haya añadido a la base de datos.

Se ha elegido guardar un archivo por firma por que se ha considerado que así se aumenta fiabilidad del sistema.

```
typedef struct firma{
    int dx [Muestreo];
    int dy [Muestreo];
    int y [Muestreo];
    int p [Muestreo];
    int numero_puntos;
}firma;
```

Figura 3.9: Struct firma, modelo de datos utilizado para manejar las firmas

Otras opciones que se estudiaron para guardar de las firmas fueron la copia punto a punto de los N puntos de la firma, pero por razones de eficiencia (menor tiempo de lectura y escritura) se eligió hacer el copiado de la estructura entera. De esta forma no son necesarios bucles.

Ademas el copiado de la estructura entera evita la lectura fácil del archivo por terceros. Al ser una estructura y no estar codificado con una codificación de texto, por ejemplo unicode UTF-8 o ACII, no se puede abrir con un editor de texto aunque tenga extensión *".txt"*.

El método **"estado base de datos"** trabaja con los archivos de la base datos. Comprueba cuantas firmas verdaderas y falsas tiene cada usuario y también cuantos usuarios hay. Se ejecuta cada vez que se va a hacer un barrido de la base de datos, y cuando se añaden usuarios o firmas. De esta forma evitamos que el programa este atado a unos parámetros fijos. Un usuario puede tener solo 3 firmas almacenadas y seguir funcionando bien el programa, aunque baje la fiabilidad del mismo.

### 3.2.4. DTW

El algoritmo DTW es el que lleva el peso computacional del sistema. En este apartado se explica como se ha implementado le algoritmo. Dentro de este bloque hay dos métodos generales, uno diseñado para la detección de un usuario con una

firma de entrada (1 to all) y otro que esta diseñado para el ajuste y entrenamiento del sistema.

En la figura 3.10 se pueden ver las partes que componen el bloque DTW. También se ven los flujos (flechas) de información que viajan de unos métodos a otros. Las flechas negras son los flujos de entrada y los azules los de salida. Las funciones (en óvalos) se localizan en los diferentes archivos, dtw.c y dtw_functions.c (rectángulos amarillos).

La división en estas funciones hace que se puedan intercambiar los parámetros del DTW, ver el apartado 2.3.1, sin necesidad de modificar el programa general, ya que, con cambiar el método correspondiente es suficiente. De esta forma se puede cambiar la restricción de pendiente o el tipo de coste local fácilmente.

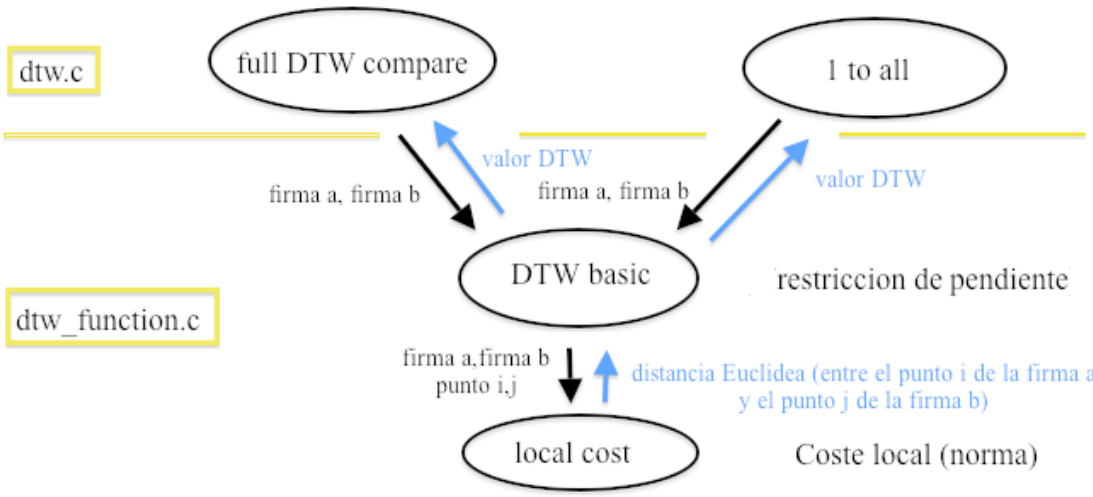



Figura 3.10: Flujo de datos entre los métodos de DTW

- **Full DTW compare:** diseñado evaluar el sistema (calcular el error). Compara todas las firmas verdaderas con todas (verdaderas y falsas) que se encuentran en la base de datos. Tiene una característica muy importante ya que guarda en un array el conjunto de valores DTW fruto de la comparación y en otro array la etiqueta cada valor de DTW segun que firmas ha comparado con el siguiente código:

| | | |
|---|---|---|
| misma firma | 0 | |
| 2 verdaderas del mismo usuario | 1 | *usuario licito* |
| 2 verdaderas de distintos usuarios | 3 | *imitador casual* |
| 1 verdadera y 1 falsa de distinto usuario | 3 | *imitador casual* |
| 1 verdadera y 1 falsa del mismo usuario | 2 | $falsificador$ (*imitador*) |

Este etiquetado es muy útil a la hora de calcular el error FAR causal, FAR imitador, FAR global y FRR. Se hablará de esto más detalladamente en el apartado de clasificación.

- **Comparación 1:N (1 to all):** Este método sirve para la detección de usuario, compara la firma de entrada con todas las firmas verdaderas (5 por usuario) de la base de datos. Para evitar errores, hay un método que guarda el estado de la base de datos, así en la comparación no se intenta acceder a firmas que no existen.

  Dado que el número de firmas es flexible, hay un array que guarda el numero de usuario de cada comparación DTW, para que la detección sea más rápida.

- **DTW basic:** Ambas funciones nombradas anteriormente llaman a este método. Es la restricción de pendiente. Las características ya se han explicado en el apartado anterior 3.1.4.

- **Coste local (Norma):**calcula la distancia Euclidea entre 2 puntos de las firmas que le llegan por parámetro. Tiene en cuenta las 4 variables (dx, dy, y, p). Este método solo es llamado por la función DTW basic.

### 3.2.5. Clasificador

Una vez conseguido el valor del ajuste por el algoritmo DTW de todas las parejas de firmas posibles se necesita evaluar el resultado para su correcto interpretación.

El clasificador diseñado basa su elección en un umbral dado. Si el valor de DTW es menor que el umbral entonces se consideran que las 2 realizaciones son del mismo usuario, si por el contrario es mayor, la firma no corresponde al mismo usuario. Tras probar el sistema y corroborar las conclusiones expuestas , se ha visto que para conseguir resultados buenos es necesario tener 5 firmas por usuario para la comparación por lo que la detección es un poco más compleja.

- **Detecta usuario:** es utilizado en el modo identificación del programa. El método tiene como entrada dos arrays. Uno de los valores DTW resultado de la comparación de la firma de entrada con todas las firmas verdaderas de la base de datos y otro con el numero de usuario al que corresponde ese valor DTW. La forma de funcionamiento es la sigiente:

  1. Hace la media aritmétca de de los 5 resultados DTW de comparar la firma de entrada con las 5 firmas cada usuario. De esta forma se tiene un valor DTW por cada usuario.
  2. Si un solo valor medio es inferior al umbral, el valor de usuario es el resultado
  3. Si por lo contrario hay varios usuarios cuya media DTW es menor que el umbral se escogerá el que menor valor DTW tenga.

- **Calcula error:** sirve para entrenar y evaluar el sistema. Tras clasificar las firmas por aceptadas o no según un umbral dado, calcula los siguientes porcentajes:

  1. Proporción de falsos rechazados (FRR)
  2. Proporción de falsos aceptados entrenados (imitaciones) (FAR skilled)
  3. Proporción de falsos aceptados casuales (FAR casual)

4. Proporción de falsos aceptados global (FAR global)

- **Calcula umbral:** Esta es una de las funciones más importantes que ha hecho que el sistema funcione. Gracias a ella se ha llegado a un umbral global válido para todas las comparaciónes. Lo que hace este método es calcular el umbral óptimo. Barre desde el valor DTW mínimo (que siempre es '0', fruto de comparar 2 firmas iguales) hasta el máximo. En este barrido se calcula el error en cada iteración con la función anteriormente nombrada "Calcula error".

  En la figura 3.11 se explica como se ajusta el umbral para minimizar el error. Se calculan FAR skilled y FRR con cada valor de umbral, el umbral se va incrementado. El punto donde ambas curvas se cruzan es el punto EER (Equal Error Rate), punto donde ambos errores son iguales.

  El otro error, FAR casual, no entra en la optimización ya que es mucho menor que el de las imitaciones (FAR skilled) y si se optimiza el de imitaciones el casual se mantiene muy bajo.

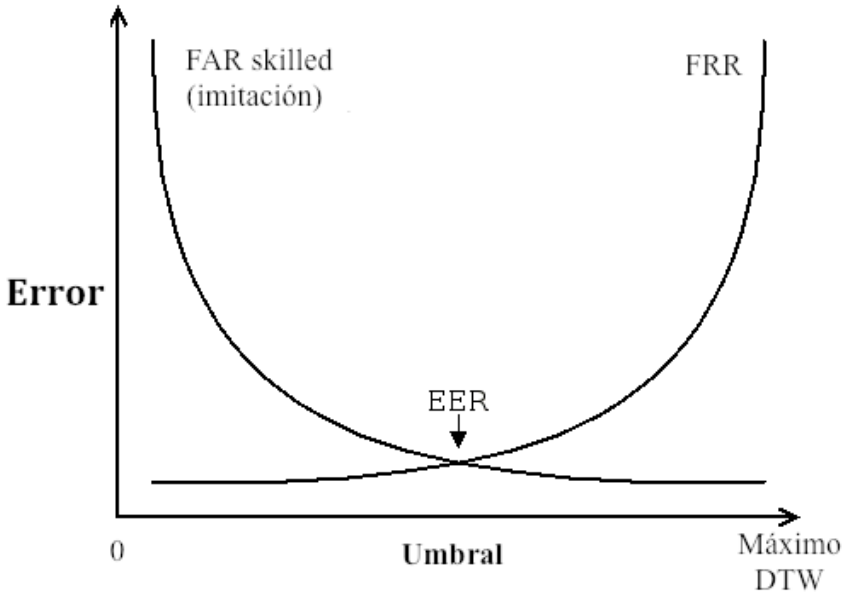



Figura 3.11: Calculo del umbral óptimo, punto EER (Equal Error Rate)

Finalmente el umbral óptimo para el sistema es el que marca el punto EER en la gráfica.

Este ajuste es general para cualquier usuario.Si la aplicación se utiliza en el futuro para fines de alta seguridad, se podría estudiar la posibilidad de implementar un umbral para cada usuario [9].

## 3.3. Problemas e imprevistos

En esta sección se explicarán las dificultades que se han tendío al desarrollar el sistema. El problema que más tiempo ha llevado en encontrar su solución ha sido leer los datos de entrada de la tableta Wacom. También se ha solucionado un problema con el calculo de error que se detalla posteriormente.

### 3.3.1. Wacom drivers

Este apartado explicará los diferentes pasos que se seguido para poder leer los datos de entrada de la tableta Wacom con un programa en C. La lectura de los datos de entrada desde la tableta Wacom se realiza por medio del puerto USB. Se comenzará con la parte de desarrollo en Ubuntu, despues las diferentes pruebas con la Raspberry Pi y finalmente se explicará la solución encontrada.

#### Desarrollo del driver en Ubuntu

La primera fase de implementación del sistema se centró en la captura de los datos de la wacom, ya que se consideraba que era la parte más difícil por su falta de documentación. Esta parte se realizó con un ordenador portatil con Ubuntu instalado.

Se implementó un driver siguiendo las lineas de los drivers oficiales de Wacom para poder leer la información de la conexión USB. En esta implementación se utilizaron la biblioteca de desarrollador **libusb** diseñadas para C (libusb-1.0-dev). De esta biblioteca se puede destacar las siguientes funciones utilizadas:

- **libusb_get_device_descriptor:** sirve para obtener la información de un dispositivo USB. Se utiliza para identificar la tableta por su fabricante 'idVendor'y su número de producto 'idProduct'.
- **libusb_open:** sirve para abrir la comunicacion Input/Output con el dispositivo.
- **libusb_detach_kernel_driver:** utilizada para desactivar cualquier driver que este activo en el sistema para este dispositivo. Esto es necesario ya que solo un driver puede leer la información de entrada.
- **libusb_interrupt_transfer:** Utilizada para copiar la información desde el endpoint (explicado en el apartado 2.5) a una espacio reservado de memoria por el programa.

#### Pruebas del driver en Raspberry Pi

Tras desarrollar el driver en un ordenador con Ubuntu y un procesador X86, se probó en la Raspberry Pi con Raspbian (Linux Debian modificado) instalado. Para ello se compiló el driver, en la misma versión de compilador (gcc-4.6) para evitar errores y con las librerias libusb-1.0-dev correctamente instaladas en el Raspberry Pi.

Los resultados de la prueba fueron fallidos, pues el driver no leía los datos de la tableta wacom. La tableta era detectada correctamente, por lo tanto, parte de las funciones que forman parte de la libusb funcionaban correctamente. El

problema se detecta cuando no se copian los datos del "endpoint" a la variable correspondiente.

En este punto se estudiaron diferentes razones por las cual no funcionaba el driver en Raspbian. Aquí se enumeran los diferentes pasos para intentar localizar la razón por la cual no funciona el driver:

- **Problemas del sistema operativo:** El sistema operativo instalado en el Raspberry Pi es Raspbian, un Debian modificado. Por el contrario el sistema operativo donde se realizó y funcionó el driver es Ubuntu. Se intentó instalar Ubuntu en la Raspberry Pi pero este software solo esta disponible para el modelo 2 de la Raspberry Pi (en este proyecto se utiliza el modelo 1).

  Finalmente se probó el programa en Ubuntu sobre una Raspberry Pi 2 y no funcionaba.

- **Falta de modulos en el sistema operativo:** otra razón por la que se supuso que no funcionaba el sistema era por que, al tener instalado un Debian modificado y reducido, tal vez, los desarolladores habían eliminado un módulo necesario para el sencillo funcionamiento del driver.

  Se busco información sobre el problema, finalmente se contactó con Wacom. La empresa mostró mucho interés por le proyecto y nos dió acceso a la plataforma SDK (Software Development Kit) de Wacom. Esta plataforma nos dio acceso a 3 tipos de drivers [19]:

  1. Kernel Driver: necesario para inicializar la tableta y transformar protocolos hardware propios de Wacom en eventos USB de entrada estándar (interrupt events). Es un parche que se instala sobre el kernel de linux. Con este driver la tableta (en teoría) se puede usar como un ratón convencional.
  2. X Driver: se utiliza para transformar eventos del kernel en eventos XInput. Estos eventos XInput son usados por diferentes aplicaciones para leer datos de entrada.
  3. Libwacom: sirve para dar información sobre cualquier tableta que este conectada. Con esta biblioteca se puede configurar distintas opciones desde el panel de control para las distintas tabletas conectadas.

Para instalar el Kernel Driver hay que tener previamente los linux-headers correspondiente a la versión del kernel.Los linux-headers son archivos necesarios para compilar drivers. Estos archivos contienen, entre otras cosas, nombres de funciones que hacen llamadas al sistema.

Raspbian viene por defecto con el kernel 3.18. Estos linux-headers para esta version son "trunk" o en desarrollo (en azulen la figura 3.12). Para evitar problemas se cambió el kernel al 3.12 que tenia unos headers estables (en amarillo en la figura 3.12).

Tras cambiar el kernel de Raspbian al 3.12 se instaló el kernel driver de Wacom pero la tableta seguía sin funcionar. El driver desarrollado no leía los datos y tampoco funcionaba en modo ratón, pese a que la información de Wacom garantiza este funcionamiento en modo ratón con este driver.

```
pi@raspberrypi ~ $ apt-cache search 'linux-headers'
linux-headers-3.10-3-all - All header files for Linux 3.10 (meta-package)
linux-headers-3.10-3-all-armhf - All header files for Linux 3.10 (meta-package)
linux-headers-3.10-3-common - Common header files for Linux 3.10-3
linux-headers-3.10-3-rpi - Header files for Linux 3.10-3-rpi
linux-headers-3.12-1-all - All header files for Linux 3.12 (meta-package)
linux-headers-3.12-1-all-armhf - All header files for Linux 3.12 (meta-package)
linux-headers-3.12-1-common - Common header files for Linux 3.12-1
linux-headers-3.12-1-rpi - Header files for Linux 3.12-1-rpi
linux-headers-3.18.0-trunk-all - All header files for Linux 3.18 (meta-package)
linux-headers-3.18.0-trunk-all-armhf - All header files for Linux 3.18 (meta-package)
linux-headers-3.18.0-trunk-common - Common header files for Linux 3.18.0-trunk
linux-headers-3.18.0-trunk-rpi - Header files for Linux 3.18.0-trunk-rpi
linux-headers-3.18.0-trunk-rpi2 - Header files for Linux 3.18.0-trunk-rpi2
linux-headers-3.2.0-4-all - All header files for Linux 3.2 (meta-package)
linux-headers-3.2.0-4-all-armhf - All header files for Linux 3.2 (meta-package)
linux-headers-3.2.0-4-common - Common header files for Linux 3.2.0-4
linux-headers-3.2.0-4-rpi - Header files for Linux 3.2.0-4-rpi
linux-headers-3.6-trunk-all - All header files for Linux 3.6 (meta-package)
linux-headers-3.6-trunk-all-armhf - All header files for Linux 3.6 (meta-package)
linux-headers-3.6-trunk-common - Common header files for Linux 3.6-trunk
linux-headers-3.6-trunk-rpi - Header files for Linux 3.6-trunk-rpi
linux-headers-rpi - Header files for Linux rpi configuration (meta-package)
```

Figura 3.12: linux headers disponibles

Una vez instalado el primer driver de Wacom (kernel driver) y no conseguir resultados, se intaló el segundo driver de wacom, (X-Driver) para ver su comportamiento.

Para instalar el X-Driver de Wacom hay que tener previamente en el sistema el X-server. Este es un sistema de ventanas muy típico de los entornos Unix tal como Linux o Mac. Programas como Wireshark necesitan este entorno para funcionar.

También se necesitó instalar las herramientas de "automake" y "autoconf" de los repositiorios para poder añadir el sistema de X-server a Raspbian. Finalmente se instaló el X-Driver de Wacom y la tableta empezá funcionar como ratón.

No se instaló la tercera biblioteca de Wacom (libwacom) ya que después de ver el código solamente servía para tener las características de las tabletas en un programa y además, la Wacom Inutos 4 PTK-440 , no estaba en la lista, por lo que esta biblioteca no ayudaría en la lectura de los datos.

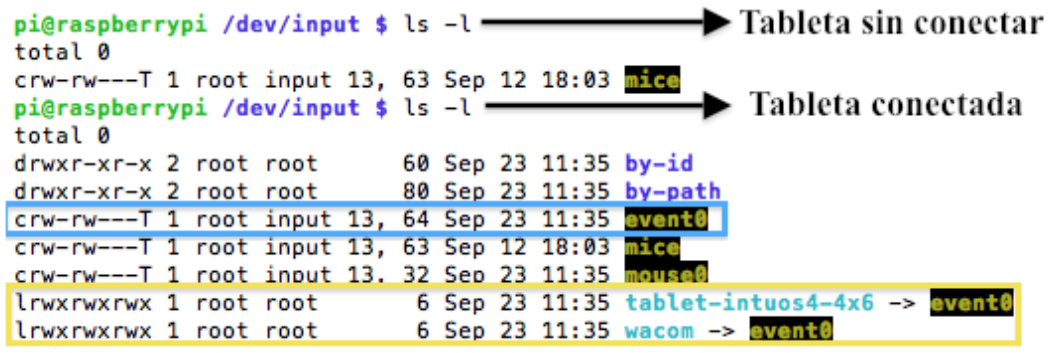



Figura 3.13: ejecutar el comando "ls-l" en el directorio /dev/input con y sin la tableta conectada

### Solución final

Se explicará la solución a la que se llegó tras no conseguir que el driver desarrollado en Ubuntu funcionara en la Raspberry Pi. Como ya se explicó en el apartado anterior, tras instalar todos los drivers recomendados por Wacom, solo se consiguió la funcionalidad de ratón.

Debido a que la tableta funcionaba como ratón, se buscó el directorio donde se localiza el buffer que los dipositivos conectados (como los ratones) utilizan para comunicarse con el sistema. Este directorio es el */dev/input*.

En la figura 3.13 se puede ver los diferentes archivos creados en el directorio */dev/input* al conectar la tableta Wacom. El buffer de entrada del USB es el *event0*, señalado en Azul. Señalado en amarillo, *tablet-intuos4-4x6* y *wacom* son diferentes enlaces simbólicos a el buffer de entrada *event0*. El sistema también ha creado un dispositivo de ratón, *mouse0* (la tableta funciona como ratón con el X-Driver instalado).

Al leer el buffer de entrada de la wacom, *event0* con la orden *cat*, obtenemos valores que no están codificados para su lectura, no se entienden. Estos valores están visibles en la figura 3.14

```
pi@raspberrypi /dev/input $ cat event0
??V?
    ?/??V?
          ^6??V?
                ??V?
                    E??V?
                         V?
                           ??V?
                               @??V?
                                    ???V?
                                         ??V??
```

Figura 3.14: leyendo el buffer event0 con el comando cat

Se buscó otras opciones para adquirir la información desde un buffer de entrada y que los datos fueran legibles. Como ya se ha comentado brevemente en el apartado 3.2.1, se utilizó un módulo llamado **usbmon** del paquete **debugfs**.

El módulo usbmon es un analizador de paquetes para la conexión USB. Existen otros analizadores de paquetes más conocidos para otras conexiónes, tcpdump es el más extendido, diseñado para conexiones de red. Usbmon es para usb lo que tcpdump es para la conexión de red.

Gracias al módulo usbmon se puede leer el buffer de entrada USB del sistema. Los datos que se obtienen en este buffer es facilmente legible, apta para la identificación de paquetes y la información que estos contienen.

```
pi@raspberrypi /dev/input $ sudo ls /sys/kernel/debug/usb/usbmon
0s  0u  1s  1t  1u
```

Figura 3.15: buses de lectura disponibles

El usbmon tiene un socket general llamado 0u (color amarillo de la figura 3.15). En este socket se junta la información de todos los buses usb. El número de buses USB depende del hardware. La Raspberry Pi solo tiene un bus USB, llamado 1u

(color azul en la figura 3.15). Los 4 puertos visibles en la figura 2.17 son creados por medio de un hub usb.

Como se puede ver en la figura 3.16, la información en este buffer (1u) es más legible que la lectura mostrada anteriormente del buffer (event 0) 3.14. Esta es la fuente de información que se ha utilizado para obtener los datos de la tableta Wacom. Los datos que se leen no son solo información de la wacom. También hay tramas de control y tramas de comunicación con otros dipositivos. Por este motivo se ha tenido que implementar un filtro para leer solo las tramas que corresponden al sensor de la wacom. La identificación de las mismas se explica en el apartado 3.2.1.

```
^Cpi@raspberrypi /dev/input $ sudo cat /sys/kernel/debug/usb/usbmon/1u
da383a60 1243403652 S Ci:1:003:0 s c0 a1 0000 0114 0004 4 <
da383a60 1243403775 C Ci:1:003:0 0 4 = 40080000
da383a60 1243403817 S Co:1:003:0 s 40 a0 0000 0114 0004 4 = 41080000
da383a60 1243403934 C Co:1:003:0 0 4 >
da383a60 1243403981 S Ci:1:003:0 s c0 a1 0000 0114 0004 4 <
da383a60 1243404122 C Ci:1:003:0 0 4 = 40080000
da383ae0 1243404478 S Bo:1:003:2 -115 174 = a6300000 a6400000 22003200 58b035ad 0f32b827 eb5eb248 08004510 00940ee8
da383ae0 1243404572 C Bo:1:003:2 0 174 >
da383a60 1243404675 S Ci:1:003:0 s c0 a1 0000 0118 0004 4 <
da383a60 1243404808 C Ci:1:003:0 0 4 = 2d780000
da246220 1243405284 C Bi:1:003:1 0 78 = 20004800 7a75b827 eb5eb248 58b035ad 0f320800 45100034 c95f0000 40062bf1
```

Figura 3.16: una captura del buffer de entrada 1u con el comando cat

Ahora se explica cual es el proceso para tener acceso a estos buffers. Esta herramienta no está activa por defecto en ningún sistema Linux.

Para ejecutar esta herramienta (usbmon) necesitamos montar los directorios de debugfs. Para ello se ha creado un script que se ejecuta al iniciar el sistema operativo. La primera linea del script monta los directorios de debugfs en */sys/kernel/debug*. La segunda linea muestra los dispositivos USB conectados. Esto sirve para conseguir el número de dispositivo que el sistema asigna a la tableta y que el programa necesita para su lectura, más detalles en el apartado 3.2.1.

```
sudo mount -t debugfs none_debugs /sys/kernel/debug
sudo cat /sys/kernel/debug/usb/devices
```

```
T:  Bus=01 Lev=02 Prnt=02 Port=00 Cnt=01 Dev#=  3 Spd=480  MxCh= 0
D:  Ver= 2.00 Cls=ff(vend.) Sub=00 Prot=01 MxPS=64 #Cfgs=  1
P:  Vendor=0424 ProdID=ec00 Rev= 2.00
C:* #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=  2mA
I:* If#= 0 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=ff Driver=smsc95xx
E:  Ad=81(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=83(I) Atr=03(Int.) MxPS=  16 Ivl=1ms

T:  Bus=01 Lev=02 Prnt=02 Port=03 Cnt=02 Dev#= 12 Spd=12   MxCh= 0
D:  Ver= 1.10 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=056a ProdID=00b8 Rev= 1.04
S:  Manufacturer=Tablet
S:  Product=PTK-440
C:* #Ifs= 1 Cfg#= 1 Atr=80 MxPwr=300mA
I:* If#= 0 Alt= 0 #EPs= 1 Cls=03(HID  ) Sub=01 Prot=02 Driver=wacom
E:  Ad=81(I) Atr=03(Int.) MxPS=  10 Ivl=4ms
```

Figura 3.17: Captura de los dispositivos USB conectados con el comando “cat /sys/kernel/debug/usb/devices”.

En la figura 3.17 se pueden ver toda la información disponible sobre los dispositivos USB. En nuestro caso se tiene que buscar el numero del dispositivo "Dev#" de la tableta conectada ( Wacom Intuos 4 PTK-440). La forma de reconocerlo puede ser bien por le nombre (señalado en azul) o por el Vendor y Produc ID , señalado en amarillo.

### 3.3.2. Alta tasa de error

Durante en desarrollo del prototipo han surgido problemas tanto con la medida del error y como con su valor elevado. A continuación se explicará como ha evolucionado el sistema desde el cálculo y la tasa de error inicial al los valores y ajustes finales.

#### Calculo de un falso punto EER

Al principio del desarollo (una vez llegado el momento de calcular el error de clasificación), se obtenían unas tasas de error en torno al 5 %. Este error era el punto EER (previamente explicado en el apartado 3.2.5) pero con una ligera variación. La curva FAR no correspondía con FAR skilled (firmas imitadas) que se puede ver en la figura 3.11 sino que era una ponderación de los errores FAR casual y FAR skilled. El error FAR(general) se calculaba siguiendo esta fórmula:

$$FAR_{general} = \frac{(Firmas\ aceptadas\ casuales + Firmas\ aceptadas\ imitadas)}{Numero\ total\ firmas\ casuales + imitadas} \times 100 \tag{3.7}$$

En la ecuación 3.7 se explica como se calculaba el error FAR(general). Las *firmas aceptadas casuales* son las firmas correspondientes a otro usuario (o a la imitación de otro usuario) que han pasado el umbral de aceptación.. El valor *firmas aceptadas imitadas* es el número de firmas imitadas que se han dado por validas. El denominador es el número todal de firmas casuales e imitadas que tenemos en nuestra base de datos.

La razón por la que no es válida la optmización del sistema por medio de la curva FAR(general)-FRR es que el valor de FAR skilled es muy elevado. Para dar un ejemplo, con un FAR(general) de 2 % el FAR skilled se situa en torno a 20 % siendo el FAR casual del 2 %.

La razón de que el FAR skilled tenga tan poca importancia en el valor de FAR(general), es que el número de firmas casuales era mucho mayor que el de firmas falsificadas. Para 1 usuario, se estaban comprobando 5 firmas falsas y 990 firmas casuales.

Posteriormente se cambió la búsquedad de error minimo al punto EER de la curva FAR skilled FRR. Se comprobó que el FAR casual siempre muy por debajo del skilled.

En la gráfica 3.18 se puede ver el cambio que implica buscar un umbral respecto a la curva FAR casual (azul) que valdría 162, el error EER valdría 3 % y tendría un error de FAR skilled de 20 %, a buscar un umbral respecto FAR skilled donde el punto EER se situa en 7 % y FAR casual esta por debajo de 1 %.

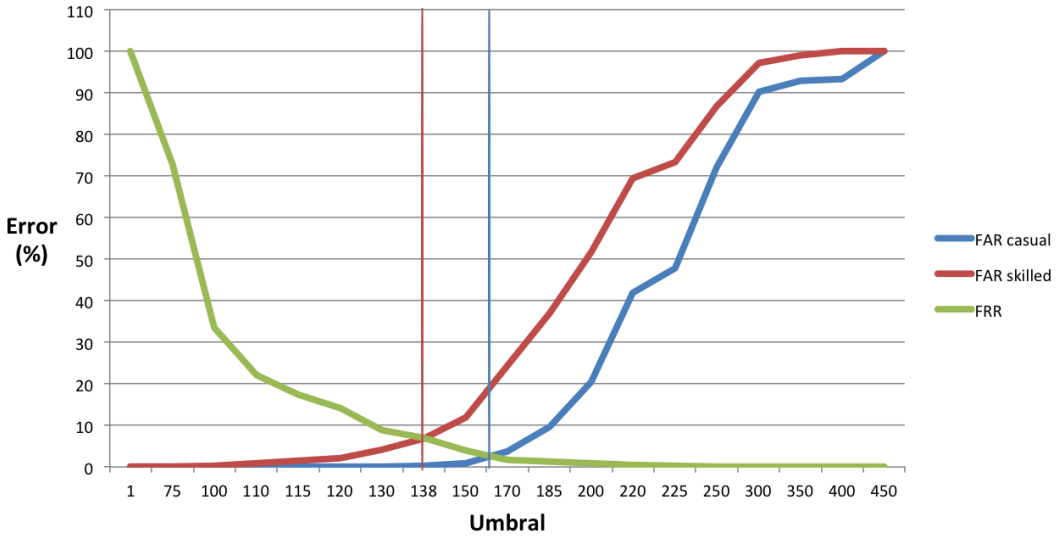



Figura 3.18: Calculo del umbral óptimo en la comparación

### Media DTW de las 5 firmas

Como se puede ver en la gráfica 3.18, el error EER optimizado respecto a la curva FAR skilled es del 7 %, demasiado elevado par aun sistema de identificación. Para reducir este error se puso en práctica el concepto de variabilidad de la firma. Con este método se esperaba reducir sustancialmente el error hasta niveles donde otros estudios habían llegado.

Para reducir el error se hizo la media de los valores DTW entre la firma de entrada y las 5 firmas que pertenecen al usuario de la base de datos. Más detalles de la base de datos en 3.1.3.

$$DTW_{firma_A,usuario_N} = \frac{1}{5}\sum_{i=1}^{5} DTW_{firma_A,firma_i} \tag{3.8}$$

En la ecuación 3.8 se explica como se realiza la media aritmética de los valores DTW entre 1 firma de entrada *firma A* y las 5 firmas *firma i* del *usuario N* que están almacenadas en la base de datos.

A continuación se muestra la gŕafica 3.19 de las curvas de FAR skilled, FRR. Para conseguir estos datos se ha hecho la media DTW explicada anteriormente. El conjunto de datos es el mismo que en la gráfica 3.18.

Como conclusión la utilización de las 5 firmas en la clasificación reduce el error del 7 % a 4 % en este conjunto de 100 usuarios (250 firmas verdaderas y 250 falsas). Probando este método sobre otros conjuntos de datos, la reducción es similar.

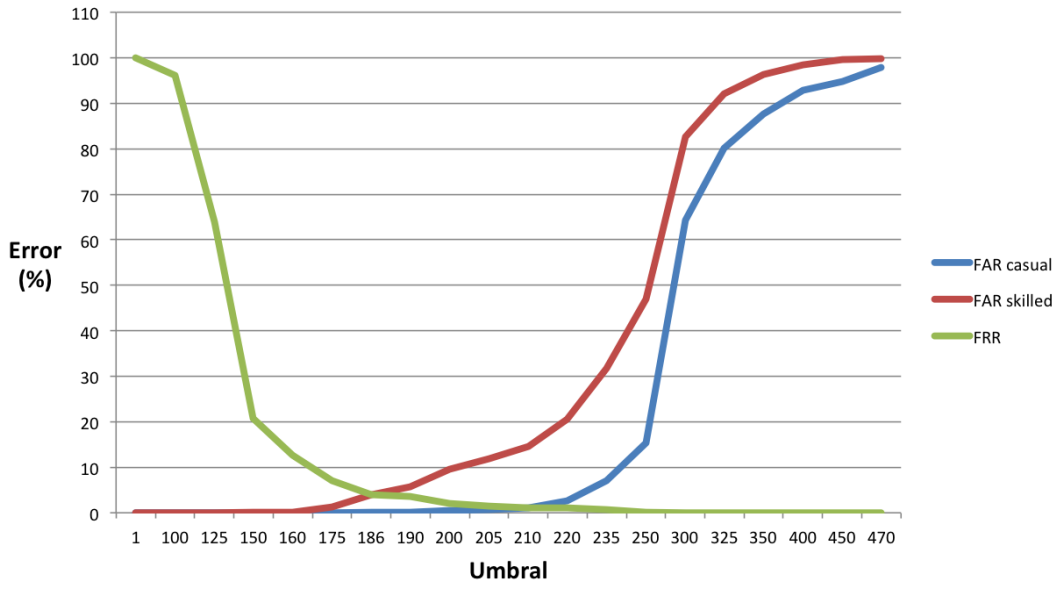



Figura 3.19: Calculo del umbral óptimo en la comparación realizando la media dtw entre la firma de entrada y las 5 firmas del usuario.

## 3.4. Ajuste del sistema, entrenamiento

En este apartado se explica los pasos que se han dado para ajustar los diferentes parámetros del sistema asi como los métodos que se han utilizado para su entrenamiento.

### 3.4.1. Muestreo general

Se explicará las razones por las cuales se han interpolado todas las firmas igualando el número de puntos para su posterior comparación.

#### Variabilidad de longitud en las firmas

La longitud de las firmas tiene una gran varianza. La firmas varían desde 47 puntos (medio segundo de realización) a 1500 muestras (15 segundos de realización).Estos datos se han obtenido utilizado la base de datos MCyT.

Esta gran diferencia de puntos hace que necesitemos una interpolación para que todas firmas tenga la misma longitud y se puedan compara con el algoritmo DTW.

El error de detección aumenta cuanta más diferencia haya en las firmas. Esta afirmarción se respalda de varias pruebas hechas al respecto sobre un subconjunto de la base de datos MCyT.

#### Definición de la prueba

Se marcó un muestreo maximo (30,50,100,200,300). Todas las firmas definidas con menos puntos que el muestreo máximo no se interpolaban (dejando su longitud sin variar). Las firmas que superaban en número de puntos al muestreo

máximo se inerpolaban linealmente. Todas las firmas se normalizan geométricamente respecto su centro de gravedad. El error (punto EER) se calcula por método explicado en el apartado 3.2.5, (buscando el umbral óptimo).

Se han elegido estos valores pues la firma que menos puntos tiene son 47. Se eligió un valor por debajo de este(30) y un valor muy cercano (50). despues para ver la progresion de la función de error se evaluaron 100, 200 y 300.

En la figura 3.20 se puede ver la evolución de esta prueba. Con muestreos menores que 50 , el error aumenta ( se pierde demasiada información en la interpolación) y con valores por encima de la longitud mínima de firma el error crece con un carácter lineal.

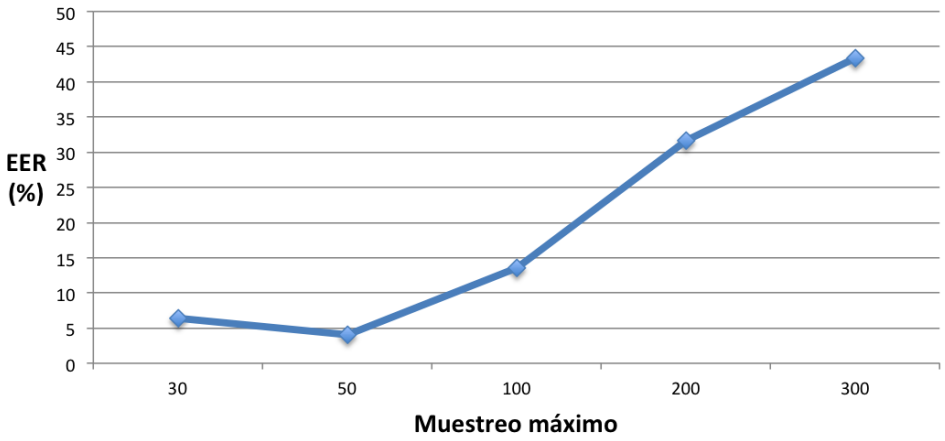



Figura 3.20: Gráfica de error EER con distintos valores de interpolación.

La conclusión es que el algoritmo DTW funciona mejor si las secuencias de datos comparadas tiene la misma longitud.

### Limitación computacional

El procesador donde este programa se ejecuta es muy limitado. Tiene una velocidad de reloj de 700 MHz. El sistema tiene que funcionar en un tiempo relativamente bajo, para que la espera del usuario para saber el resultado sea mínimo.

Si utilizamos un muestreo elevado, tiempo de computacional también aumenta ya que son directamente proporcionales. La comparación que realiza el sistema es de 1 firma (firma de entrada) con todas las de la base de datos (número de usuarios * 5 firmas cada usuario). Este relación es lineal, pero el número de puntos de la firma le añade un caracter exponencial al coste computacional. Esta forma exponencial se ve reducida en parte por la utilización de limitaciones como la ventana de Sakoe, detallada en el apartado 2.3.1. Por esta razón la figura 3.21 parece lineal, pero tiene cierto caracter exponencial.

Hay que tener en cuenta que para calcular el error y ajustar el sistema, se tiene que hacer una comparación mucho más costosa. En este caso, el coste computacional no aumenta linealemente respecto del número de firmas, sino exponencialmente. La razón es que hay que comparar todas las firmas entre ellas. Los tiempos moestrados a continuación son de identificación de un usuario ( 1:N) no de calcular el error (N:N).

### Definición de la prueba

Se ha hecho una prueba para determinar cual el es muestreo máximo nos permite el procesador. Dado que es un sistema de reconocimiento de firma, se esta buscando un valor que no produzca mucho retardo. Así conseguiremos que el sistema no sea rechazado por el usuario.

En esta prueba todas las firmas han sido interpoladas al mismo número de puntos ( 50,100,200,300,400) y se han utilizado 100 usuarios (con 5 firmas verdaderas cada uno).

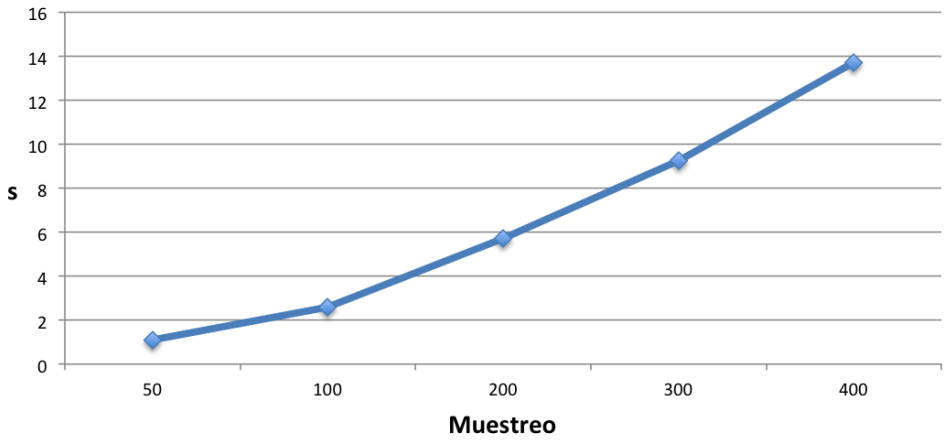



Figura 3.21: Tiempo que tarda el algoritmo DTW en hacer la comparacion 1 firma de entrada con 100 usuarios (1:500 firmas)

En la figura 3.21, se puede ver el tiempo que tarda el algoritmo DTW en procesar la comparación 1:500 firmas con diferentes longitudes de firma.

El tiempo máximo que se le ha puesto al sistema para que devuelva los resultados son 4 segundos. Se ha elegido este máximo por la relación que tiene con el tiempo que se tarda en realizar la firma. El usuario medio tarda 3,5 segundos en realizar la firma (dato obtenido de la base de datos MCyT). El tiempo que tienes que esperar es cercano al que has invertido.

Con este tiempo, se puede tener un muestreo de 150 puntos por firma en una base de datos de 100 usuarios.

En la figura 3.22 se muestra cuanto tarda el sistema en realizar la coparación de 1 firma con 1 usuario. Ya que el usuario tiene 5 firmas registradas, la comparación es 1 a 5 firmas. También podemos fiajr el valor de muestreo en función de los usuarios que tengamos registrados en la base de datos. Gracias a esta gráfica podemos calcular el número máximo de usuarios respecto de un muestreo.

En la figura 3.23 se ha fijado el tiempo de computación en 4 segundos para calcular el **número maximo de usuarios** que pueden estar registrados en la base de datos. Dependiendo de la aplicación que vaya a tener el sistema se puede elegir un muestreo optimo u otro.

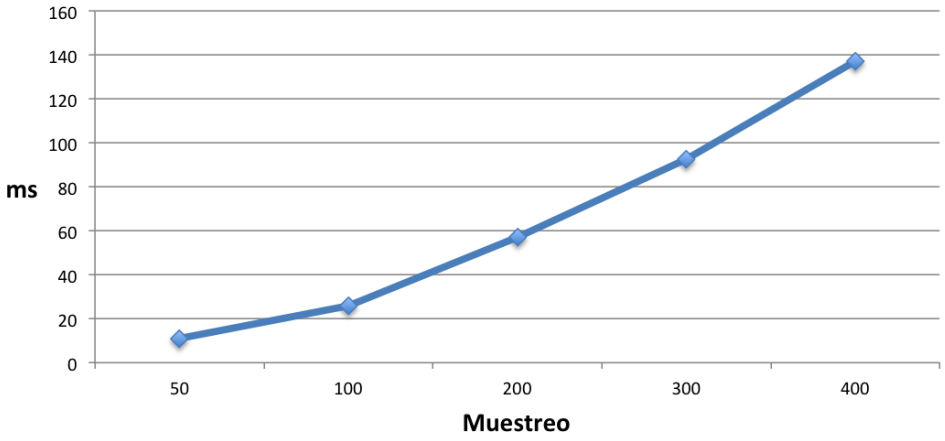



Figura 3.22: Tiempo que tarda el algoritmo DTW en hacer la comparacion 1 firma de entrada con 1 usuario. (1:5 firmas)

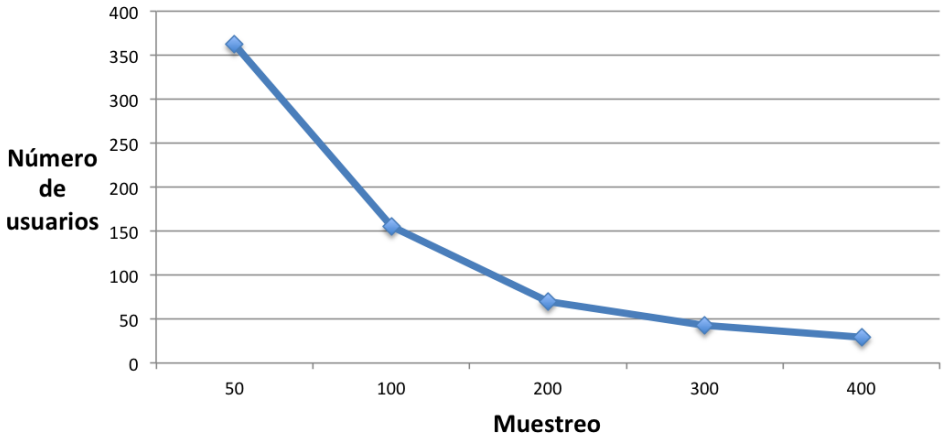



Figura 3.23: Número de usuarios máximos fijando el tiempo de computación a 4 segundos y variando el muestreo de las firmas.

### Error en función del muestreo elegido

En este apartado se explica que muestreo se ha elegido. Esta elección se ha basado en la información adquirida con las anteriores pruebas respresentadas en los gráficos. También se ha realizado otra prueba que se explicará a continuación.

Como punto de partida y para resumir el apartado anterior. El número de muestreo de las firmas tiene que ser el mismo. Todas las firmas tiene que tener el **mismo número de puntos**. Se llegó a esta conclusión tras analizar el gráfico 3.20.

El número de usuarios registrados es un valor que varía según la aplicación que tenga este dispositivo. Para cubrir un amplio rango de numero de usuarios registrados sin que aumente el tiempo de procesado se han diseñado **2 preprocesados diferentes**.

Cada preprocesado esta optimizado para un valor de muestreo. Uno de ellos funciona bien con un valor de muestreo muy bajo y el otro funciona bien cuanto

mayor es el muestreo Ambos preprocesados consiguen un error menor del 5 %.

Esta división de dos muestreos se realiza debido a la condición que no tiene que superar los 4 segundos calculando el resultado. Si se está buscando el menor error posible sin tener en cuenta el tiempo de espera para calcular el resultado, hay que elegir el primer preprocesado (explicado a continuación)

- **Preprocesado 1, óptimo para bases de datos menores de 50 usuarios**

  Este preprocesado se implementó tras haber comprobado que todas las firmas tiene que tener la misma longitud para el buen funcionamiento del algoritmo. Y comprobar el mal funcionamiento del preprocesado 2 con muestreos altos. Todos los pasos seguídos corresponden con los mismos utilizados en el trabajo [9].

  Aqui se enumeran (por orden) las transformaciónes que sufren los datos:

  1. Normalización geométrica por centro de gravedad:
  2. Interpolación lineal
  3. Normalización estadística: Normaliza la media y varianza de cada característica.

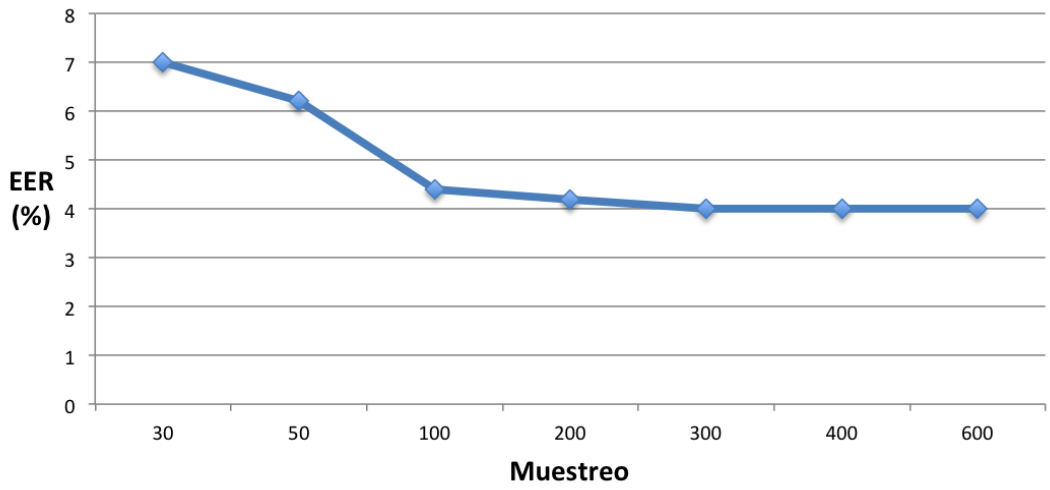



Figura 3.24: Error del preprocesado 1 en función de el muestreo

Como se puede ver en la figura 3.24 el error disminuye cuanto más muestras se toman en la interpolación. El muestreo optimo se encuentra en los **300 puntos**, ya que hacia la derecha la gráfica es plana, el error apenas se reduce.

La ocasión para utilizar este muestreo es cuando nuestra base de datos no va a superar los **50 usuarios**. De esta forma el tiempo de computación estaria en torno a 4 segundos.

- **Preprocesado 2, óptimo para base de datos mayores de 50 usuarios.**

  Se diseñó en un primer momento. Al ver que ante muestreos altos,el error aumentaba,se empezó a implementar el otro preprocesado. Debido al funcionamiento de este preprocesaod se llegó a la conclusión que el algoritmo DTW funciona mejor con longitudes de firma iguales.

El funcionamiento del preprocesado es más sencilla que el anterior. Sigue los siguientes pasos:

1. Normalización geométrica por centro de gravedad.

2. Interpolación lineal: Es una interpolación lineal que solo modifica las firmas que tienen más puntos que el muestreo elegido. Las firmas que tienen menos puntos no se interpolan.

En la figura 3.25 se puede ver como el error es minimo en el punto 50. Esto es debido a que la firma más corta que ha evaluado este sistema tiene 47 puntos. Cuando se aumenta el muesteo el error aumenta ya que las firmas con menos muestras no se interpolan ( se mantienen con el mismo número de puntos) por lo que como ya se explicó anteriormente, el DTW se degrada.

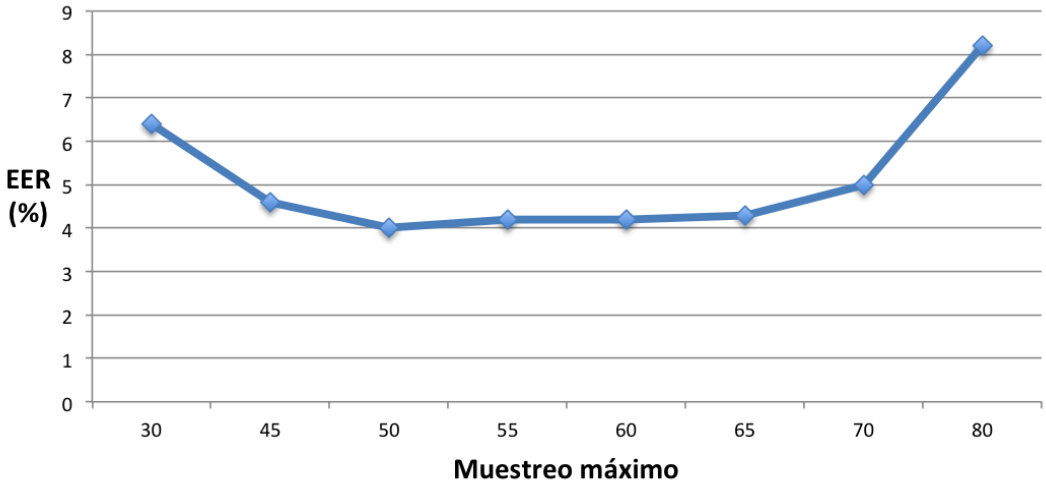



Figura 3.25:

El **muestreo óptimo** para este preprocesado esta cercano al número de puntos de la firma más corta. En este caso 50 puntos.

Este método se ha probado para otro subconjunto de la base de datos cuya firma más corta se componía de 60 puntos y el resultado es similar trasladado 10 puntos hacia la derecha.

El preprocesado 2 (con 50 puntos de muestreo) puede utilizarse en bases de datos de hasta 350 usuarios para mantener el tiempo computacional máximo de 4 segundos. Para base de datos mayores habría que estudiar otras opciones.

#### Conclusión sobre el muestreo y elección de preprocesado

Se ha elegido el preprocesado 2 con un muestreo de 50 puntos por que nuestra base de datos actualmente es de 101 usuarios.

Este preprocesado tiene menos error que el preprocesado 1. Si se elige el preprocesado 1 con 101 usuarios se tendría que ajustar el muestreo a 160 puntos (siguiendo la gráfica 3.23).Con ese muestreo en la segun la gráfica 3.24, el sistema tiene un error por encima del 4 %. Mientras que el preprocesado 2 es del 4 %.

Otra razón es que la comparación es mucho más rápida (en torno a 2 segundos) y que se puede incrementar hasta 350 usuarios manteniendo el tiempo de espera bajo.

### 3.4.2. Umbral

Como ya se explicó en el apartado 3.2.5, el sistema tiene un umbral prefijado que es el encargado de decidir si la firma de entrada corresponde con algún usuario de la base de datos o no corresponde.

Este umbral se ha ajustado buscando un balance entre firmas verdaderas rechazadas (FRR) y firmas falsas (imitaciones) aceptadas (FAR skilled). El óptimo es donde ambos errores son iguales (o muy cercanos).

El umbral óptimo se ha calculado con un conjunto de firmas y se ha probado en otro conjunto para calcular su error. EL valor umbral dtw es de 186.

Los errores conseguidos son de 3,2 % y 4 %.

### 3.4.3. Mejoras

Se han pensando diferentes formas para reducir tanto el tiempo de computación como el error. Estas ideas han sido implementadas y se ha visto que mejoran el rendimiento del general.

#### Reducción tiempo de ejecución

El principial problema de procesar muchas firmas es que el tiempo que el algoritmo tarda aumenta. Para calcular el valor hay que elegir el numero de usuarios y el muestreo y multiplicarlo por el tiempo por usuario visible en el gráfico 3.22.

Para reducir este tiempos se ha diseñado un filtro que decidir si la pareja de firmas necesita calcular su valor DTW o no es necesario. Este filtro tiene como parámetro de entrada la longitud en puntos de la firma ( el tiempo de realización). Se ha puesto el umbral siguiendo la variabilidad que tiene las realizaciones de la misma firma. El conjunto estudiado para ajustar este umbral es el MCyT.

El filtro detecta si 2 firmas no son del mismo usuario por medio de su longitud. Si la diferencia entre su longitud es mayor que el umbral lo descarta automáticamente asignandole un valor DTW muy alto (muy por encima del umbral de clasificación).

```
if(400<abs((firma_a->numero_puntos)-(firma_b->numero_puntos))){
        (*dtw)[counter]=401.0;
}else{
        (*dtw)[counter]=dtw_basic (firma_a,firma_b);
}
```

El umbral (400) se ha calculado segun un estudio realizado a la base de datos MCyT donde la máxima variabilidad puntos entre 2 realizaciónes de una misma firma es de 398 puntos (casi 4 segundos).

En el código se calcula el valor absoluto de la resta de las longitudes de las firmas. Posteriormente se compara con el umbral (400) y si es mayor, se asigna el valor DTW=401,0. Si es menor se calcula el valor DTW con la función DTW_basic explicada en el aparado 3.2.4.

Con este filtro se reducen las firmas comparadas un 6 %. Este valor se ha comprobado con 2 subconjuntos de la base de datos MCyT.

### Reducción del error

Durante la implementación y las pruebas se observó que los primeros valores de velocidad ("dx" y "dy") entre las firmas verdaderas y las imitaciones era muy diferentes. Se intentó utilizar esta diferencia para que la clasificación funcionara mejor y el error se redujera.

A continuación se explica como se ha añadido esta mejora al sistema, así como la prueba que se realizó para comprobar su efectividad.

Se implementó una variante en la función de coste local, explicado en el apartado 3.2.4 que originalmente era la distancia euclídea. Esta variante de coste local modifica la distancia euclídea de los primeros puntos de la firma. La modificación produce un incremento en el resultado final de la alineación DTW entre las dos firmas.

$$C_{ij}^{AB} = \sqrt{(y_i^A - y_j^B)^2 + (p_i^A - p_j^B)^2 + multiplicador \times ((dy_i^A - dy_j^B)^2 + (dx_i^A - dx_j^B)^2)} \tag{3.9}$$

En la ecuación 3.9 se ve la modificación que se ha hecho a la distancia euclídea. Los valores "A" y "B" corresponden con las dos firmas que se están comparando. La variable "i" es el número de punto de la firma "A" y "j" corresponde al número de punto dentro de la firma "B". "y" es la coordendada Y, "p" es la presión, "dy" y "dx" son las velocidades Y y X respectivamente.

La variable "multiplicador" es el encargado de incrementar en caso necesario el valor de las velocidades. Si "multiplicador" = 1, entonces la distancia es euclídea, si el multiplicador = 6 entonces la distancia es mucho mayor.

Este "multiplicador" es el que le da más peso a las velocidades en el valor final del alineamiento DTW.

Esta modificación solo esta activa en un cierto conjunto de valores. La característica que se esta midiendo es la velocidad de "arranque", que es la velocidad con que el usuario comienza a realizar la firma. Por esta razón solo los puntos inicales son los que tiene que sufrir la modificación.

En la figura 3.26 se ve la banda (en azul) que corresponde con el conjunto de valores que son modificados. Esta se extiende desde i=0 hasta i=3. y desde j=0 hasta j=3.

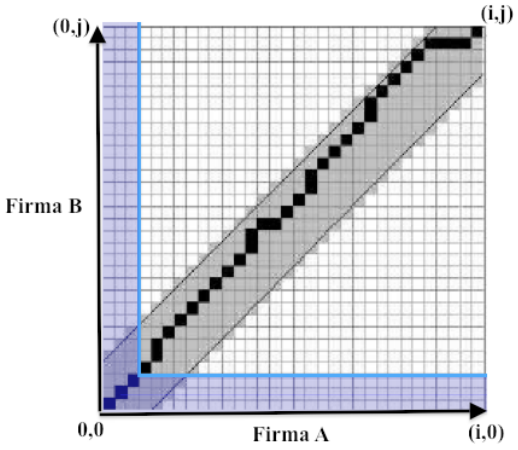



Figura 3.26: En azul esta el domínio donde la mejora esta activa.

A continuación se muestra el código que implementa esta modificación. Si el punto evaluado esta dentro de la condición, el multiplicador se iguala a 6. Si el punto cae fuera, la distancia calculada es la euclídea.

```
if((i<3)||(j<3)){
        multiplicador=6;
}else{
        multiplicador=1;
}
```

### Definición de la prueba

A continuación se explica como se fijaron los valores de anchura de la banda y del multiplicador. Para darle valided a estos valores, se calcularon para un conjunto de firmas y se evaluaron en otro conjunto.

### Búsqueda de la anchura de la banda

Primero se hizo una prueba para decidir cuantos números debían incluirse en esta modificación (anchura de la banda en la figura 3.26). Para ello se eligió un multiplicador arbitrario (multiplicador= 2) y se buscó el de error mínimo.

En la figura 3.27 se ve el error (EER) obtenido con diferentes número de puntos evaluados. El valor "0" corresponde con el error original (sin esta mejora). El valor "1" significa que la distancia euclídea del primer punto fue modificada. El valor "2" significa que la distancia euclídea de los 2 primeros puntos fue modificada y así sucesivamente.

Como conclusión, la modificación de la distancia euclídea de los **primeros 3 puntos** es la óptima.

El conjunto evaluado son 100 usuarios con 5 firmas verdaderas y 5 firmas falsas (imitaciones) cada uno, un total de 500 firmas. El muestreo utilizado fueron 41 puntos. Por esta razón el erro es más alto que en gráficas mostradas con anterioridad.

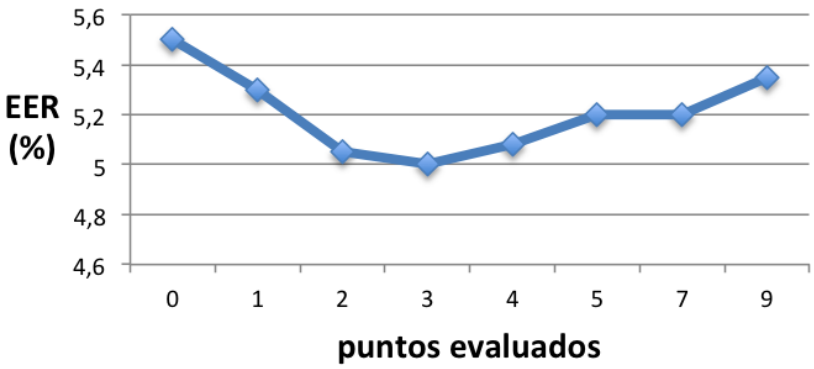



Figura 3.27: Búsqueda del error minimo varando los puntos evaluados (primeros N puntos de las firmas)

### Búsqueda de multiplicador

Una vez fijado el número de puntos evaluados se buscó el multiplicador óptimo. Para buscar el valor se hizo una prueba muy parecida a la anterior.

Esta prueba se realizó con el mismo conjunto de datos. Esta vez se fijó el número de puntos que que sufrían la modificación. Se elgió el valor 3 de acuerdo con la conclusión del ajuste anterior.

En la figura 3.28 se puede ver el error obtenido con diferentes valores de multiplicador. El error es mínimo cuando el multiplicador es igual a 6.

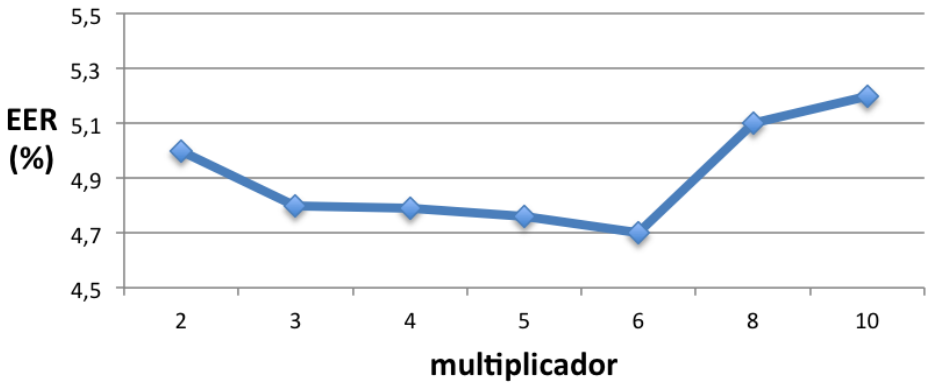



Figura 3.28: Evaluación de error según diferentes valores de multiplicador. La banda es de los 3 primeros puntos.

### Valores finales

Para resumir, los valores óptimos de esta mejora son:

- Modificar la distancia euclídea de los 3 primeros puntos de la firma
- En esta modificación multiplicar por 6 las velocidades

Con esta implementación el error (EER) se reduce del 5,5 al 4,7. Se ha probado esta mejora (con los mismos ajustes) en otro conjunto de firmas y el error también se reduce una proporción similar.

Se consideran estos ajustes como buenos ya que aunque el error ha disminuido en los 2 conjuntos. Además se han calculado los ajustes óptimos en el segundo conjunto y han resultado ser los mismos que los detallados anteriormente; Evaluar los 3 primeros puntos y multiplicar la diferencia de velocidades por 6.

## 3.5. Funcionamiento e Interfaz de usuario

En la realización de este prototipo se ha intentado cuidar todos los aspectos tanto a la hora de implementarlo como de presentarlo. Para que sea más sencillo de cara al usuario, se ha diseñado una sencilla interfaz de usuario.

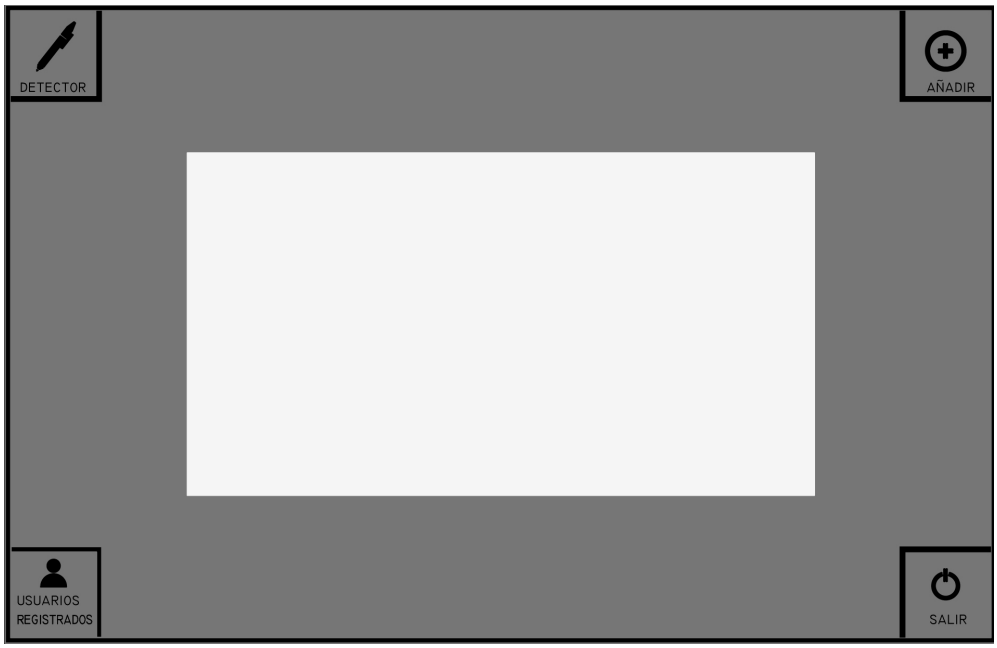



Figura 3.29: Interfaz diseñada para la tableta Wacom. Corresponde con el area activa de la tableta.

El programa se controla mediante la tableta donde se firma. En la figura 3.29 se muestra el area sensible de la tableta. Cada esquina tiene un botón con una función asignada. Estos botones se accionan pulsando con el bolígrafo de la tableta.

- La esquina inferior izquierda sirve para informarte de cuantos usuarios hay registrados en la base de datos.

- La esquina superior izquierda sirve para identificar un usuario. Una vez elegida esta opción el ususario tendrá que realizar la firma para su detección. Una vez procesada la firma, se mostrará por pantalla el número de usuario que corresponde o un aviso de firma inválida en caso de que no corresponda con ningún usuario.

- La esquina superior derecha sirve para registrar un usuario. Se informará por pantalla de los pasos a seguir. El usuario tendrá que realizar 5 firmas para que su registro sea efectivo.
- La esquina inferior derecha sirve para salir del sistema.

## 3.6. Evaluación y resultados finales

En este apartado se mostrarán las características del la última implementación del sistema. Los parámetros que finalmente se han elegido y su comportamiento. A continuación se resumen las aspectos por los cuales este sistema se caracteriza:

- Se ha elegido el preprocesado 2 con un muestreo de 50 puntos. Este preprocesado se caracteriza por realizar una normalización geométrica por centro de masas y una interpolación lineal solamente a las firmas que tengas más puntos que los definidos en el muestreo, dejando a las demás con el mismo numero de puntos.
- También se ha implementado el filtro para reducir las firmas compradas por el DTW. Este filtro analiza la longitud de las firmas y si supera un umbral prefijado no se realiza la comparación DTW.(Reduce un 6 % el tiempo de ejecución).
- Se ha reducido el error por medio del análisis de la velocidad de arranque de la firma, velocidad de los primeros puntos de la realización, (de un 5,5 a 4,7 %)
- El algoritmo DTW funciona con una restricción de pendiende P(0) asimetrica. El camino que toma DTW en la optimización esta limitado por la ventana de Sakoe.
- Por ultimo el valor DTW que tiene una firma de entrada al comparase con un usuario (que tiene 5 firmas) es la media de los resultados DTW que resultan de las 5 comparaciones. Este método de utilizar 5 firmas reduce el error de un 7 % a un 4 %.

Con esta elección se consigue un tiempo de comparación 1:1 firmas de 2,2 ms. Este tiempo de procesado también incluye la copia de una de las firmas desde la tarjeta SD.

Para tener una referencia, el tiempo que tarda en hacer la misma operación un ordenador con un Intel core 2 duo a 2,26GHz con un disco duro SSD es de 0,18ms (12 veces menos).

| Procesador | Disco duro | Tiempo (ms) |
|---|---|---|
| Rpi ARM 11, 700Mhz | Tarjeta SD | 2,2 |
| Intel core 2 duo, 2,26 GHz | SSD (estado solido) | 0,18 |

Figura 3.30: Comparativa de tiempos entre Raspberry Pi (ARM11) y Macbook pro (intel core 2 duo)

El error por el cual el sistema se caracteriza es del 4 % .Se ha obtenido entrenando el umbral de decisión para un conjunto y evaluando el error este umbral tiene en otro conjunto diferente.

# Capítulo 4

# Presupuesto

En el siguiente apartado se detalla una estimación de los diferentes costes originados durante la realización de este trabajo de fin de grado. En esta estimación se han tenido en cuenta los costes del personal, de software y de hardware utilizados durante el desarrollo del mismo. Basándose en la plantilla proporcionada por la Universidad Carlos III de Madrid se hace un desglose del coste total para facilitar la comprensión del mismo. De esta forma, se explica en los distintos apartados primero los costes del personal, seguido de los costes de software y hardware y finalizando el apartado con el coste total. Las unidades de cada tabla están expresadas en euros tomando dos decimales en cada cantidad. Todo el proceso ha sido realizado digitalmente sin necesidad de realizar impresión de material. Por lo tanto, los costes de materiales fungibles no serán añadidos. A continuación se entra en detalle de cada parte del presupuesto.

## 4.1. Coste del personal imputable al proyecto

Para calcular el coste del personal que ha trabajado en el proyecto se detallan las horas dedicadas por Pablo Díez López, autor del proyecto, y por Luis Mengíbar Pozo, tutor del mismo. El coste por horas ha sido definido basándose en la plantilla proporcionada por la universidad, teniendo en cuenta que el valor de un Ingeniero Senior es 36 Euros/hora y de un Ingeniero Junior 22 Euros/hora. Se ha supuesto un esfuerzo constante en una media de 5 horas diarias, y una dedicación de 20 días cada mes, para el caso del autor. Para el caso del tutor, se ha considerado una participación de 14 horas divididas en 10 tutorías. A partir de esos datos se realiza la figura 4.1.

| Apellidos y nombre | Categoría | Coste Euros/hora | Horas Totales | Coste total |
|---|---|---|---|---|
| Díez López, Pablo | Ingeniero Junior | 22 | 617 | 13.574,00 |
| Mengíbar Pozo, Luis | Ingeniero Senior | 36 | 14 | 504,00 |
| | | | **Total** | **14.078,00** |

Figura 4.1: Coste del personal detallado

## 4.2. Coste de Software y Hardware

La siguiente tabla resume los dispositivos y herramientas que han sido utilizados en el desarrollo del trabajo. Los software empleados, (Raspbian, Wacom SDK, Texshop y Bibdesk), se añaden a la tabla a pesar de ser productos gratuitos. Para cada elemento se nombra el producto, su precio, su periodo de amortización y el coste que supone en el proyecto. Todos los dispositivos se han usado desde el principio del trabajo y durante 6 meses, excepto el ordenador Fujitsu Siemens que solo ha sido utilizado en los últimos 4 meses del proceso, para la implementación del sistema.

## 4.3. Elementos de Hardware

En este apartado se explican cada herramienta utilizada y sus características principales.

### 4.3.1. Macbook Pro 2009

El ordenador empleado para el desarrollo del trabajo es un Macbook Pro de 13'3 pulgadas. Toda la documentación ha sido explicada mediante los software TexShop para el lenguaje Latex y Bibdesk para la bibliografía. Sus características son las siguientes:

- Procesador: Intel Core 2 Duo a 2,26GHz
- Tarjeta gráfica: Nvidia Geforce 9000 M con 256 MB DDR3 compartidos
- RAM: 4 GB DDR3 a 1033MHz
- Sistema operativo: Snow leopard
- Disdo duro : 250 GB SSD (Samsung 840 EVO)

### 4.3.2. Fujitsu Siemens

Los últimos cuatro meses del proyecto se utilizó el ordenador Fujitsu Siemens debido a que el sistema operativo instalado era Ubuntu, muy similar al del dispositivo del proyecto. Las ventajas frente al Macbook eran: similitud de compilador y bibliotecas de C con el Raspberry Pi. Sus características son las siguientes:

- Procesador: Intel Core 2 Duo a 1,6GHz
- Tarjeta gráfica: ATI Mobility Radeon HD 2400 con 256 MB dedicados
- RAM: 2GB DDR2 a 667MHz
- Sistema operativo: Ubuntu 14.04 LTS
- Disdo duro : 160 GB HDD

### 4.3.3. Raspberry pi 1 model B+

Raspberry Pi es un ordenador de placa reducida o (placa única) (SBC) de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas. Este dispositivo no incluye disco duro ni unidad de almacenamiento, por lo que se ha comprado una tarjeta SD sandisk de 16 GB. Tampoco incluye una fuente de alimentación ni carcasa. Sus características son las siguientes:

- Tipo de procesador: ARM1176JZF-S
- Tarjeta gráfica: Broadcom VideoCore IV a 250 MHz
- Velocidad del procesador: 700 MHz
- Memoria RAM: 512MB SDRAM a 400MHz
- Sistema operativo: Linux Raspbian

### 4.3.4. Wacom Intuos 4 ptk 440

Para el reconocimiento de firmas, para todo el trabajo gráfico y las pruebas de éstas, se ha utilizado la tableta gráfica Wacom Intuos 4 ptk 440. A esta, se le ha acoplado un folio, para que las características de las nuevas firmas tomadas sean lo más parecido posible a las firmas reales de la base de datos, teniendo en cuenta que la textura del bolígrafo sobre el papel es diferente a la del bolígrafo sobre la tableta. Las propiedades de este dispositivo son las descritas seguidamente:

- Tecnología: EMR (Electro de resonancia magnética)
- Área activa: 157.5 X 98.4 mm
- Niveles de presión: 2048
- Resolución: 200 lpmm (5080 lpi)
- Exatitud (lápiz): +/-0.25 mm
- Rango de inclinación (todos los lápices): 60 grados
- Software: Wacom SDK

### 4.3.5. Kingston 4 GB

Para el almacenamiento de toda la información, base de datos, documentación gráfica y de las diferentes copias de seguridad se ha utilizado una memoria USB Kingston de 4GB de capacidad y dimensiones 55.65mm x 17.3mm x 9.05mm.

## 4.4. Resumen de costes

A continuación se muestra el presupuesto total alcanzado. En la primera tabla (figura 4.2) se resumen los costes descritos anteriormente y seguidamente se les añade una tasa de costes indirectos del 20 %. A este resultado se incluye en la segunda tabla (figura 4.3) un coste del 18 % en concepto de Impuestos sobre el Valor Añadido.

| Presupuesto Costes Totales | **Presupuesto Costes Totales** |
|---|---|
| Personal | 14.078 |
| Amortización | 184 |
| Subcontratación de tareas | 0 |
| Costes de funcionamiento | 0 |
| Costes Indirectos | 2.852 |
| **Total** | **17.114** |

Figura 4.2: Coste total sin IVA

| Presupuesto Costes Totales | **Presupuesto Costes Totales** |
|---|---|
| Total sin IVA | 17.114 |
| IVA 18% | 3.081 |
| **TOTAL** | **20.195** |

Figura 4.3: Coste total con IVA

**UNIVERSIDAD CARLOS III DE MADRID**
**Escuela Politécnica Superior**

**PRESUPUESTO DE PROYECTO**

1.- Autor:
Pablo Díez López
2.- Departamento:
Departamento Electrónica
3.- Descripción del Proyecto:
- Titulo: **Sistema Empotrado para el reconocimiento de firma manuscrita**
- Duración (meses): **6**
Tasa de costes Indirectos: **20%**

4.- Presupuesto total del Proyecto (valores en Euros):
20.195,00 Euros

5.- Desglose presupuestario (costes directos)

**PERSONAL**

| Apellidos y nombre | Categoría | Coste Euros/hora | Horas Totales | Coste total |
|---|---|---|---|---|
| Díez López, Pablo | Ingeniero Junior | 22 | 617 | 13.574,00 |
| Mengíbar Pozo, Luis | Ingeniero Senior | 36 | 14 | 504,00 |
| | | | **Total** | **14.078,00** |

[a] 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

**EQUIPOS**

| Descripción | Coste (Euro) | % Uso dedicado proyecto | Dedicación (meses) | Periodo de depreciación | Coste imputable [d] |
|---|---|---|---|---|---|
| TexShop | 0,00 | 100 | 6 | 60 | 0,00 |
| Bibdesk | 0,00 | 100 | 6 | 60 | 0,00 |
| Wacom SDK | 0,00 | 100 | 6 | 60 | 0,00 |
| Raspbian | 0,00 | 100 | 6 | 60 | 0,00 |
| Wacom Intuos 4 ptk 440 | 220,00 | 100 | 6 | 60 | 22,00 |
| Raspberry pi 1 model B+ | 30,00 | 100 | 6 | 60 | 3,00 |
| Macbook Pro 2010 | 1.100,00 | 100 | 6 | 60 | 110,00 |
| Fujitsu Siemens | 700,00 | 100 | 4 | 60 | 46,67 |
| Kingston 4GB | 8,00 | 100 | 6 | 60 | 0,80 |
| Tarjeta SD sandisk 16 GB | 13,00 | 100 | 6 | 60 | 1,30 |
| | | | | **Total** | **183,77** |

[d] Fórmula de cálculo de la Amortización:

$$\frac{A}{B} x C x D$$

**A** = nº de meses desde la fecha de facturación en que el equipo es utilizado
**B** = periodo de depreciación (60 meses)
**C** = coste del equipo (sin IVA)
**D** = % del uso que se dedica al proyecto (habitualmente 100%)

**SUBCONTRATACIÓN DE TAREAS**

| Descripción | Empresa | Coste imputable |
|---|---|---|
| | | |
| | **Total** | 0,00 |

**OTROS COSTES DIRECTOS DEL PROYECTO[e]**

| Descripción | Empresa | Costes imputable |
|---|---|---|
| | | |
| | **Total** | 0,00 |

[e] Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes

| Presupuesto Costes Totales | Presupuesto Costes Totales |
|---|---|
| Personal | 14.078 |
| Amortización | 184 |
| Subcontratación de tareas | 0 |
| Costes de funcionamiento | 0 |
| Costes Indirectos | 2.852 |
| **Total** | **17.114** |

| Presupuesto Costes Totales | Presupuesto Costes Totales |
|---|---|
| Total sin IVA | 17.114 |
| IVA 18% | 3.081 |
| **Total** | **20.195** |

Figura 4.4: presupuesto detallado

# Capítulo 5

# Conclusiones generales

A lo largo de este documento se ha ido explicando el proceso que se ha llevado a cabo para la realización de este proyecto. Se ha estudiado el algoritmo DTW con sus diferentes parámetros, se ha tenido en cuenta las diferentes opciones de hardware y elegido el mas apropiado, tambien ha sido objeto de investigación las diferentes formas de comunicación entre la tableta y el programa C por medio del puerto USB y por ultimo se ha implementado y optimizado el sistema.

Una vez completado el desarrollo, el resultado es un identificador de usuarios por medio de firma manuscrita. El cálculo del resultado se realiza en una Raspberry Pi 1 B+ y la adquisición de la firma se realiza mediante una tableta Wacom Intuos 4 PTK 440.

Con la realización de este prototipo se ha comprobado la posibilidad de utilizar procesadores empotrados para identificación biométrica por medio de firma manuscrita. Asi como la viabilidad de utilizar este proyecto como punto de partida para el desarollo de un sistema de identificación debido a que su tasa de equierror se encuentra en el 4 % para firmas falsas entrenadas y menor del 0,2 % para firmas falsas aleatorias.

Se puede afirmar que el resultado final ha sido satisfactorio ya que el sistema desarrollado funciona con con un error bajo y el tiempo de procesamiento es menor que 2 segundo teniendo a 100 usuarios en su base da datos. Además esta base de datos se puede ampliar hasta 350 usuarios manteniendo el tiempo de procesado en torno a los 4 segundos. Debido a que este retardo es muy similar al tiempo medio que tarda un usuario en realizar la firma, no causaría rechazo.

Finalmente cabe mencionar que que se han cumplido los objetivos del proyecto:

1. Estudio e implementación del algoritmo DTW.
2. Optimización del algoritmo para el hardware elegido.
3. Realización de un prototipo identificador de firma manuscrita con un procesador empotrado.

# Capítulo 6

# Trabajo futuro

En este capítulo se expone las diferentes líneas que se puede seguir desarollando este proyecto.

- **Reducción del tiempo de computación mediante el módulo LOGI PI**. Esta FPGA ha sido especialmente diseñada para la Raspberry Pi. Este modulo se conecta por medio de los pines GPIO a la placa y tiene las mismas dimensiones que la Raspberry Pi. Se puede trasladar el calculo DTW a un modulo fpga y así reducir sustancialmente el tiempo computacional.

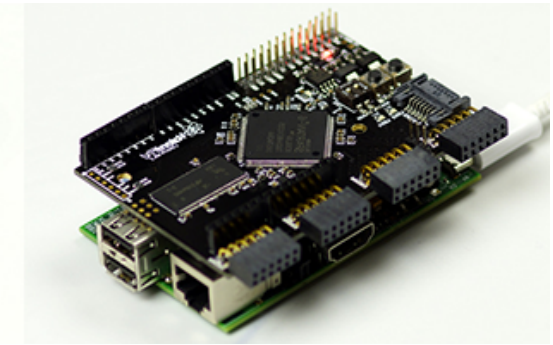


Figura 6.1: módulo LOGI PI (encima) conectado a Raspberry Pi (debajo)

- **Implementación del algoritmo en la Raspberry Pi 2 y optimizar el algoritmo para su procesador de 4 nucleos**: La realización de este algoritmo en la nueva Raspberry Pi que cuenta con un procesador de 4 núcleos abre un gran abanico de posibilidades. La mas interesante es la posibilidad de paralelización del algoritmo en estos 4 nucleos por medio de la división en subconjuntos de la base de datos. Con esta división se puede reducir el tiempo de procesamiento ya que se están haciendo comparaciones paralelamente.
- **Adaptar el sistema para una utilidad real**: Una de las trabajos mas interesante implementar el sistema en un entorno real. Debido a su base de datos máxima de 350 usuarios puede aplicarse por ejemplo para control de accesos de pequeñas empresas.
- **Implementación del algoritmo en otros hardwares:** Otra via de desarrollo es implementar el sistema en otros dispostivos detallados en el apartado

2.4. La placa Nvidia Jetson TK1 es especialemente interesante ya que ya se podría portar el sistema a este hardware haciendo la computación del DTW paralelamente en los núcleos de la GPU. Esta paralelización es factible y se puede reducir el tiempo a la mitad [3].

# Appendices

# Planificación

En este capítulo se muestra de forma escrita y gráfica la planificación del proyecto. Las diferentes fases incluyen investigación, implementación, pruebas, mejoras del sistema, elaboración de la documentación y presentación del trabajo fin de grado. Cada una de ellas se explicará en un apartado diferente, a pesar de que muchas de ellas han ido solapándose temporalmente. En esta planificación se incluyen también las reuniones con el tutor para el seguimiento del proyecto, omitiendo el tiempo empleado en la comunicación por medios electrónicos. Para su mejor entendimiento se añade un diagrama de Gannt al final del capítulo para entender con más facilidad los plazos de cada fase.

**Fases del proyecto:**

1. **Fase de investigación** La primera fase del proyecto fue la propuesta de enunciado, junto con la búsqueda de información para su desarrollo. En esta fase se produce una serie de estudios previos seguida de un análisis de la información encontrada sobre trabajos y sistemas similares de reconocimiento de firmas, así como de los diferentes algoritmos para reconocimiento de firmas. También se analizaron las opciones de hardware que había.

   -Duración: 60 días/ 300 horas.

2. **Reuniones con el tutor**

   Desde el primer día de la propuesta de enunciado y durante los 6 meses de duración del trabajo se han ido programando tutorías para corrección y resolución de dudas. Estas no han sido constantes. Al principio se realizaron tutorías mensuales de una hora de duración y en el último mes una reunión semanal de dos horas de duración. Esta fase ha ido solapándose con las demás según se muestra en el diagrama.

   -Duración: 14 horas.

3. **Fase de implementación**

   Esta es la fase más importante y la que más esfuerzo ha necesitado. Supone toda la planificación y ejecución del proyecto, incluyendo el diseño y estructura de la aplicación, así como las pruebas y mejoras del sistema. Durante esta fase se ha necesitado buscar información para la resolución de dudas así como reuniones con el tutor.

   -Duración: 40 días/ 200 horas.

(Teniendo en cuenta 25 horas a la semana, la tarea de implementación ha sido realizada en 2 meses que suponen 200 horas)

A continuación muestran las partes de la implementación:

- Diseño y estructura de la aplicación El comienzo de esta fase de implementación se produjo con el diseño de la aplicación y la estructura de todo el trabajo que habría que llevar a cabo, puesto que el trabajo completo ha sido realizado mediante pequeñas partes.

  -Duración: 2 días/ 10 horas.

- Implementación de la aplicación Supone la programación de todas estas partes o módulos que se ha dividido la aplicación. Empezando por los módulos de adquisición de datos y terminando por el clasificador.

  - Duración: 25 días/ 125 horas.

- Pruebas Una vez realizada la implementación, se procede a hacer las pruebas del sistema para comprobar su funcionamiento. Esta fase se ha repetido varias veces hasta disminuir el porcentaje de error hasta el 4 %.

  - Duración: 3 días / 15 horas.

- Mejoras del sistema Con la primera prueba el porcentaje de error era demasiado alto, por ello se han de realizar mejoras y correcciones en la programación para disminuir dicho porcentaje. Estas mejoras se han repetido tantas veces como las pruebas hasta concluir en un buen porcentaje de error.

  - Duración: 10 días/ 50 horas.

4. **Elaboración de la documentación**

   A pesar de que una parte de la memoria fue escrita antes de la implementación del proyecto, como la planificación presente y el estado de arte, la mayoría fue descrita una vez acabado el sistema. Es por ello que esta parte también puede verse solapada con las demás. Esta parte recoge todo el proceso digital que ha supuesto la realización de este trabajo.

   -Duración: 30 días/ 100 horas.

5. **Presentación**

   Una vez terminadas todas las fases y entregada la documentación, se procede a una presentación oral frente al tribunal para la exposición del trabajo fin de grado. Esta fase no se ha realizado todavía por lo que el tiempo es estimado suponiendo una duración de tres horas.

   - Duración: 3 horas

Sumando todas las horas obtenemos un total de 617 horas empleadas en este proyecto. A continuación se añade una tabla resumen de estas actividades incluyendo la fecha de inicio y fin de cada tarea, (figura 2). Un diagrama de Gannt (figura 3) cierra el capítulo, explicando gráficamente los espacios temporales que ha necesitado cada actividad.

| | Nombre Tarea | Duración | Comienzo | Fin | Predecesoras |
|---|---|---|---|---|---|
| 1 | **Reuniones con el tutor** | **10 días** | **27/03/2015** | **24/09/2015** | |
| 2 | Reunión 1 | 1 día | 27/03/2015 | 27/03/2015 | |
| 3 | Reunión 2 | 1 día | 16/04/2015 | 16/04/2015 | |
| 4 | Reunión 3 | 1 día | 07/05/2015 | 07/05/2015 | |
| 5 | Reunión 4 | 1 día | 28/05/2015 | 28/05/2015 | |
| 6 | Reunión 5 | 1 día | 25/06/2015 | 25/06/2015 | |
| 7 | Reunión 6 | 1 día | 23/07/2015 | 23/07/2015 | |
| 8 | Reunión 7 | 1 día | 03/09/2015 | 03/09/2015 | |
| 9 | Reunión 8 | 1 día | 15/09/2015 | 15/09/2015 | |
| 10 | Reunión 9 | 1 día | 17/09/2015 | 17/09/2015 | |
| 11 | Reunión 10 | 1 día | 24/09/2015 | 24/09/2015 | |
| 12 | **Fase de investigación** | **60 días** | **27/03/2015** | **26/06/2015** | |
| 13 | Propuesta del proyecto | 1 día | 27/03/2015 | 27/03/2015 | |
| 14 | Estudios previos | 35 días | 30/03/2015 | 22/05/2015 | 13 |
| 15 | Análisis información | 24 días | 25/05/2015 | 26/06/2015 | 14 |
| 16 | **Fase de implementación** | **40 días** | **29/06/2015** | **21/08/2015** | |
| 17 | Diseño y estruct de la aplicación | 2 días | 29/06/2015 | 01/07/2015 | |
| 18 | Implementación de la aplicación | 25 días | 02/07/2015 | 21/08/2015 | 17 |
| 19 | Pruebas | 3 días | 10/08/2015 | 21/08/2015 | 18 |
| 20 | Mejoras del sistema | 10 días | 10/08/2015 | 21/08/2015 | 19;18 |
| 21 | **Elaboración de la documentación** | **30 días** | **13/04/2015** | **18/09/2015** | |
| 22 | Introducción | 2 días | 24/08/2015 | 25/08/2015 | |
| 23 | Estado del Arte | 8 días | 13/04/2015 | 22/04/2015 | |
| 24 | Desarrollo del sistema | 13días | 26/08/2015 | 11/09/2015 | |
| 25 | Conclusiones | 2días | 10/09/2015 | 11/09/2015 | 24;23 |
| 26 | Planificación y presupuesto | 3 días | 14/09/2015 | 16/09/2015 | |
| 27 | Trabajo futuro | 2 día | 17/09/2015 | 18/09/2015 | |
| 28 | **Entrega del Proyecto** | **1 día** | **26/09/2015** | **26/09/2015** | 21;20 |
| 29 | **Presentación** | **1día** | **08/10/2015** | **08/10/2015** | |

Figura 2: planificación


Figura 3: diagrama de Gannt

# Interfaz sobre la tableta Wacom


Figura 4: Interfaz de usuario montada sobre la tableta Wacom

# Bibliografía


[1] Múltiples factores contribuyen al crecimiento del mercado de biometría del iris.

[2] Carmen Sánchez Ávila Alberto de Santos Sierra. Hand geometry: Biometría de mano para dispositivos hand geomtry: biometría de mano para dispositivos móviles.

[3] Luis Mengíbar Pozo Alejandor García molina. Aceleración de gpu de algoritmos de reconocimiento biométrico mediante firmas manuscrite on-line. Master's thesis, Universidad Carlos III MadridCarlos III Madrid, 2013.

[4] SAKOE; CHIBA. Dynamic programming algorithm optimization for spoken word recognition. 1978.

[5] Patricia de los Ángeles Alonso Fuertes. Análisis del algoritmo dtw para reconocimiento biométrico de personas mediante firma manuscrita on-line. Master's thesis, Universidad Carlos III Madrid, 2012.

[6] J. Fierrez, J Ortega-GArcia. *On-line signature verification. Handbook of Biome- trics*. 2007.

[7] The Linux Foundation. Raw data with usbmon.

[8] Carlos Mauricio Galvis Tranlaviña. Introducción a la biometría.

[9] Juan Manuel Pascual Gaspar. Uso de la firma manuscrita dinámica para el reconocimiento biométrico de personas en escenarios prácticos. Master's thesis, Universidad de Valladolid.

[10] Ariel Roche Director: Dr. Badih Ghattas. Árboles de decisión y series de tiempo. Master's thesis, Facultad de Ingeniería, UDELAR, 2009.

[11] https:biometrics/biometric-center-of-excellence/modalities/voice recognition. Voice recognition.

[12] josé Guadalupe Aguilar Hernandez. Autenticación de usuarios a través de biometría de tecleo. Master's thesis, Universidad Juárez Autónoma de Tabasco. Universidad Juárez Autónoma de Tabasco. Unviersidad Juárez Autónoma de Tabasco, 2010.

[13] Stan Z. Li. *Encyclopedia of Biometrics*.

[14] MQP Electronics Ltd. Usb made simple.

[15] NSTC Subcommittee on Biometrics. Hand geometry.

[16] Eduardo Álvarez Alonso Pablo Pérez San-José. Estudio sobre las tecnologías biométricas aplicadas a la seguridad ( inteco). 2011.

[17] Pavel Senin. Dynamic time warping algorithm reveiw.

[18] James Shueyen Tai and Kin Fun Li. Dynamic time warping algorithm: A hardware realization in vhdl.

[19] wacom. Sdk linux wacom.

[20] Prof. Dr. Javier Areitio Bertolín y Prof. Dra Teresa Areitio Bertolín. Análisis en torno a la tecnología biométrica para los sistemas electrónicos de identificación y autenticación. 2007.